A NOVEL HUMAN MACHINE INTERFACE FRAMEWORK FOR CONFLICT RESOLUTION

A Thesis
Presented to
The Academic Faculty

by

Jiaming Li

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology
May 2016

A NOVEL HUMAN MACHINE INTERFACE FRAMEWORK FOR CONFLICT RESOLUTION

Approved by:

Dr. George Vachtsevanos, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Ayanna M. Howard
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. John A. Buck
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Gisele Bennett
Georgia Tech Research Institute
Georgia Institute of Technology

Dr. Dimitri Mavris
School of Aerospace Engineering
Georgia Institute of Technology

Date Approved: [December 8, 2015]

To my daughter,
Muchen Maria Li

ACKNOWLEDGEMENTS

First of all, I would like to thank my advisor, Dr. George Vachtsevanos, who accepted me as a student of him in 2012 and financially supported me continuously. He has continued to provide unstinting academic, professional and personal guidance throughout the Ph.D. program. He showed me different ways to approach a research problem and the need to be focused and persistent to accomplish any goal.

I would also like to thank my dissertation committee members, Dr. Ayanna M. Howard, Dr. Gisele Bennett, Dr. John A. Buck and Dr. Dimitri Mavris. The special attention from Dr. Howard and Dr. Bennett in helping me refine my thesis is greatly appreciated.

A big thanks to all the members and friends of the Intelligent Control Systems Laboratory of Georgia Tech who have greatly aided me in this research, especially to Dr. Ioannis Raptis, Dr. Chris Sconyers, Ms. Honglei Li, Ms. Maggie Garven, Dr. Xuerong Ye, Dr. Daisuke Matsuura, Dr. Cheng He and Mr. June Cho.

The support and collaboration of Impact Technologies LLC, Spectro Scientific, Simtech and Analatom is acknowledged and appreciated, especially the collaborative efforts of the following people involved with these institutions: Dr. Xiang Li, Dr. Chan Hian Leng Ian, Dr. Le Tung, Dr. Pat Henning, Dr. Javier Echauz, Dr. Doug Brown. Special thanks goes to Dr. Javier Echauz for helping me to build up the right way to do projects and guide me how to develop better code.

Also, I would like to thank my friends, Fangzhou Liu, Rui Fan, Chong Han, Liangyi Sun, Zhenyu Tan, Bai Cui, Ke Li, Yu Liu, Dongqing Li, Yi Du, Dongbo Zhao, Renke Huang, Bo Zhao, Xianglei Liu, Zhe Guang, Fan Shi, Xinping Tian, and all my brothers and sisters from my church, without whom I can't successfully accomplish this task and have a good memory.

Last I want to thank the support from my family. My parents, Dr. Shengli Li and Lixia Jia, keep supporting me both financially and spiritually, which is a solid foundation of my doctoral life. And my parents in law, Jiazhu Si and Lingling Bi, special thanks to their help with the affairs in my home and taking care of my daughter during the final stage. My loving wife, Wen Si, who is so patient to endure my occasionally bad temper, keeps supporting me when I'm down and encourages me to successfully finish all the tasks. Also to my daughter, Muchen, who is right now only several months old. She brought me a lot of fun and peace during the last and hardest period.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

m	Mass value
Pl	Plausibility
Pl	Belief
FMECA	Failure Modes and Effects Criticality Analysis
HUMS	Health and Usage Monitoring Systems
CI	Condition Indicator
RUL	Remaining Useful Life
TSA	Time Synchronous Averaging
AS	Automated System
HMI	Human Machine Interface
DCBR	Dynamic Case Based Reasoning
SC	Similarity of components
RAO	Ratio of ascending order
MEB	Mean Error Bar
GUI	Graphical User Interface

SUMMARY

There is an urgent need to improve the autonomy, safety, survivability and availability of such critical assets as aircraft and robotic (unmanned) systems that are subjected to internal and/or external threats in the execution of a mission. The automated systems and the human operator are invariably exposed to different evidences that result in conflict or disagreement as to the "best" action required to remedy an emergency situation. This thesis addresses these concerns and introduces a holistic and verifiable framework for the harmonious symbiosis of a complex machine and the human operator. It takes advantage of novel enabling technologies from health management, Dempster-Shafer theory, game theory and a reasoning paradigm called Dynamic Case Based Reasoning (DCBR).

The constituent modules of the human-machine interface architecture pursued in this thesis include an on-board automated system that provides to the human operator the most accurate and reliable information regarding the platform's current and future health state through key performance metrics specific to the vehicle and onboard sensors. These are presented to the operator in a prioritized manner based on mission essential elements. A game-theoretic optimization algorithm is employed to combine conflicting and incomplete information.

The proposed human-machine interface architecture is illustrated in the thesis. The pilot observes current environmental conditions, reads the on-board displays, and communicates with the knowledge base. Similarly, the AS gathers information from the

available on-board sensor suite, represented by the DAQ module, and takes advantage of on-board or off-board algorithms to perform data mining tasks. A “smart” knowledge base is designed and adopted to function as the repository of similar historical cases with the ability to learn and adapt new ones. Case-Base Reasoning (CBR) constitutes the main system level reasoning paradigm of the architecture while incorporating essential elements of a learning strategy. Similarity metrics and Q learning algorithms are applied here to accomplish this task.

The task of the knowledge base is to discover similarities between a new case and those stored in the case library. For this purpose, appropriate similarity metrics are defined and exploited. Criteria for the automated system and pilot are defined and applied. Here the main contribution lies in defining the suitable similarity metrics for the pilot’s current evidence and those contained in the case library; the latter may be either literal or numerical information.

The simulated Graphical User Interface (GUI) is generated in MATLAB and is fully described in the thesis. The sections related to the pilot’s actions are depicted in the pilot’s input panel section as estimated actions. As shown in the thesis, the estimated pilot’s actions are in accordance with the pilot’s input sequence. The overall human-machine interface is set in a probabilistic framework taking into account uncertainty, noise, etc. A filtering action is intended to avoid suddenly changing actions and performs the decision strategy in a relatively stable manner. The thesis concludes with test cases that demonstrate the utility of the approach.

CHAPTER 1

INTRODUCTION

1.1 Introduction

There is an urgent need to improve the autonomy, safety, survivability and availability of such critical assets as aircraft and robotic (unmanned) systems that are subjected to internal and/or external threats in the execution of a mission. It has been well documented over the past years that human error is a major cause of class A aircraft mishaps. Moreover, on-board equipment malfunctions, incipient failures and environmental stresses also contribute to aircraft accidents. There is unquestionable and documented evidence of aircraft failures that are due to a lack of correct maintenance practices. Human errors are a major contributor to aircraft mishaps. It is anticipated that such failures could be avoided if the aircraft is properly instrumented to detect incipient flight critical failures and warn the pilot of pending events that may endanger the health of the vehicle.

What is required, therefore, is a fresh look at how we address this critical issue. Human errors affect the safety and reliability of aircraft at all stages of the asset's life cycle: design, manufacturing, and daily operations. Design flaws are typically remedied via extensive simulations, testing, etc., failures caused by manufacturing

defects are partially mitigated via qualification testing at the component, subsystem and system levels. Errors induced in operations and maintenance offers an opportunity for further improvement in system development of operation.

Hypothesis:

Can we design, test and evaluate an integrated methodology for aircraft (manned and unmanned) - human (pilot, human sensor) interface that exhibits unique capabilities to take advantage of on-board or ground sensing, data mining and diagnostic/prognostic algorithms, combine them efficiently and effectively with human observations, experience/expertise to provide the human (pilot) with "best" advisories for actions to be taken in the event of potentially detrimental contingencies. The question, therefore, may be posed as follows: Can we develop a systems-based architecture that will contribute to enhanced reliance of the pilot on automated processes? Can we build a rigorous, reliable and robust human-automation interface that will contribute to improved aircraft safety? What should be the major modules of this interface? How they will be integrated into both legacy and new assets? What are the potential risks and possible mitigation strategies? We will expand on these questions in the sequel as we detail the proposed methodology.

The proposed method/Proof of Concept:

Proof of the hypothesis is based on the following theoretical, testing and developmental steps:

1. An overarching human-machine interface architecture, whose constituent modules are detailed in the sequel that may be implemented and updated in real time as new evidence becomes available. The concept may be formulated as a set of cooperating, knowledge-based subsystems: two assessors and two planning subsystems, and a pilot interface. The two assessors, Situation Assessment and System Status, determine the state of the outside world and the aircraft systems, respectively.
2. Unique feature of the human-machine interface include:
    a) An Automated System (AS) module residing on-board the aircraft for manned vehicles and on the ground station (sensor) for unmanned autonomous systems that acquires sensor data/images and performs a series of processing steps from raw data pre-processing, data mining for feature or condition indicator extraction and selection, incipient failure diagnosis and prognosis of a failing component's remaining useful life, assessment of the severity of the fault and communication to the pilot via a "smart" reasoning module of the "best" advisories for corrective action.
    b) A human (pilot) component of the situational awareness module used by the pilot to list observations derived from a variety of sources, such as satellite data, meteorological information, other aircraft communications, etc., regarding the evidence that supports actions to be taken in order to mitigate the current emergency.

c) An intelligent reasoning paradigm called Dynamic Case Based Reasoning (DCBR) that receives the Automated System's advisories and the pilot's intended actions, reasons about potential conflicts between the two and activates a conflict resolution scheme that attempts to resolve such conflicts and provides the human operator the ability to interpret automated system outputs correctly and to effectively control the decision making process.

d) A decision support module whose task is to combine evidences and resolve any potential conflicts between the pilot and the automated system. Conflicts arise between the pilot's intent/commands and automated system commands/advisories. They arise from the different perceptions of the pilot and the automated routines stemming from experience, current data and information available to the pilot and the control architecture which may differ in content, quantity and means for the expedient presentation and follow-up action. The principal task of the decision support system is, therefore, to resolve conflicts between the pilot's actions and those recommended by the automated system. The enabling technologies borrow concepts from Dempster-Shafer evidential theory and game theory.

e) A Graphical User Interface (GUI) that enables the realization of all major modules of the human-machine interface in real time and allows

for the demonstration, testing and performance evaluation of the scheme under realistic example situations.

The Target Application Domains

Figure 1. Operators on the Unmanned Aircraft

1. **Manned Aircraft/Rotorcraft** those are equipped with a Health and Usage Monitoring System (HUMS) capable of receiving data from sensors monitoring critical component and system performance with particular emphasis on the health status (incipient failures) of such components. Furthermore, aircraft usage patterns are also recorded and transmitted to HUMS. It is assumed that sensors are installed on the aircraft monitoring, recording and transmitting data to HUMS.

2. Unmanned Autonomous Systems that are capable of sensing, monitoring and processing/analyzing a variety of data/information. The paradigm here is that the vehicle crew consists of at least two experienced members: A pilot that controls and maneuvers the vehicle's flight and a human "sensor" that receives all available data/images/information and performs such functions as data pre-processing/filtering, data mining, diagnostics and prognostics. The human sensor, finally, provides to the pilot relevant advisories as to the course of a mission in the presence of adverse events.

We detail in the sequel the modules of the human-machine framework.

1.2 Human-Machine Interface: The Case for Technology Improvements

Human – machine interface concerns the interface between user and machine. It plays a key role in the effectiveness of human user system in terms of the way information is passed between man and machine. Whatever task is being carried out, this interface must be intuitive and representative of the real world. The aim is to allow the user to interact naturally, concentrating on the task at hand and not on the technicalities of operating the computer. Both the way in which the computer interprets the user's actions and vice-versa are controlled by the design of the interface.

It has been well documented over the past years that human error is a major cause of class A aircraft mishaps. Moreover, on-board equipment malfunctions, incipient failures and environmental stresses contribute to aircraft accidents. There is unquestionable and documented evidence of aircraft failures that are due to a lack of correct maintenance practices. It is anticipated that such failures could be avoided if the aircraft is properly instrumented to detect incipient flight critical failures and warn the pilot of pending events that may endanger the health of the vehicle.

Table 1: Statistical distribution of fatal accident causes

Fatal Accident causes by category (%)						
Cause	**1950s**	**1960s**	**1970s**	**1980s**	**1990 +**	**Total**
Pilot Error	27	24	17	19	21	22
Pilot Error (weather related)	5	13	10	11	15	11
Pilot Error (mechanical related)	5	3	3	3	4	4
Total pilot Error	**37**	**40**	**31**	**33**	**40**	**37**
Other Human Error	2	5	6	4	5	4
Weather	19	7	8	10	6	7
Mechanical Failure	12	13	16	13	14	13
Sabotage	3	3	6	7	6	5
Other causes	0	2	2	1	0	1
Undetermined or missing	*37*	*30*	*34*	*32*	*29*	*33*

There is an urgent need to improve the autonomy, safety, survivability and availability of such critical assets as aircraft and robotic (unmanned) systems that are subjected to internal and/or external threats in the execution of a mission. What is required is a fresh look at how we address this critical issue. Human errors affect the safety and reliability of aircraft at all stages of the asset's life cycle: design, manufacturing, and daily operations. Design flaws are typically remedied via extensive simulations, testing, etc.; failures caused by manufacturing defects are partially mitigated via qualification testing at the component, subsystem and system

levels. Errors induced in operations and maintenance offers an opportunity for further development. It is anticipated that such failures could be avoided if the aircraft is properly instrumented to detect incipient flight critical failures and warn the pilot of pending events that may endanger the health of the vehicle.

In published studies, the primary focus has been on acquiring sufficient information to conclude if pilot errors indeed constitute major causes for fatal air crashes. Researchers at NASA Ames Research Center have been developing an approach to model the behavior of discrete sensors in an aircraft during flights in order to discover atypical behavior of possible operational significance [1]. The tool, called sequenceMiner, analyzes large repositories of discrete sequences of data derived from primary sensors that record pilot actions in order to identify operationally significant events. Each flight is analyzed as a sequence of events, taking into account both the frequency of occurrence of switches and the order in which switches change values. Basic elements of the sequenceMiner could be integrated into human-automation architecture.

The Pilot's Associate program, a joint effort of the Defense Advanced Research Projects Agency and the US Air Force, managed by the Air Force's Wright Laboratory, began in February 1986 as an application demonstration for DARPA's Strategic Computing Initiative. DARPA wanted to advance the program's technology base, principally in the area of real-time, cooperating knowledge-based systems. The Air Force wanted to explore the potential of intelligent systems applications to

[1] www.datamining.arc.nasa.gov

improve the effectiveness and survivability of post- 1995 fighter aircraft. The Pilot-Vehicle Interface subsystem provides the critical connection between the pilot and the rest of the system. The interface is supposed to ensure that the system as a whole provides the information the pilot wants, when it is needed. Unfortunately, at that point in time, health and usage monitoring systems were not available to the degree required to assess the health status of the vehicle and recommend appropriate mitigation measures in the presence of a contingency. More recent studies are addressing various human-machine interface architectures to allocate appropriate functions between the human and the machine to reduce the effect of uncertainty. There is no published information available as to how, in the event of incipient failure conditions, automated advisories and the pilot's evidence/experience can be combined effectively and efficiently to avoid catastrophic events.

In real time / safety critical systems, the effectiveness of HMI becomes critical. What better example than the cockpit of a modern aircraft? Gone are the days of hundreds of discrete, mechanical instruments. The modern 'glass cockpit' comprises of two or three computer screens! Here we have an example of human lives depending on how effectively a computer system can interpret control input from a pilot, and indeed on how it relays information back to the pilot.

There are numerous examples of accidents caused by failure of a pilot to enter information into a flight computer accurately, resulting in the computer making a poor last second decision costing human life! The computer's ability to take into account

external factors such as weather, human behavior, birds etc. must be considered and evaluated in addition to whether the pilot should have any direct control over the aircraft, with examples supporting both the elimination of the pilot and the reduction of automation.

The case for new and innovative tools and methods required to develop, test and evaluate a rigorous, verifiable and robust human-machine interface is supported by numerous historical events involving aircraft mishaps, as detailed in this thesis.

1.3 Backgrounds and Motivation

Lessons Learned-Events over the past Years: Air France Flight 447: 'Damn it, we're going to crash'

Airbus's 'brilliant' aircraft design may have contributed to one of the world's worst aviation disasters and the deaths of all 228 onboard. The official report by French accident investigators echoes provisional verdicts suggesting human error. There is no doubt that at least one of AF447's pilots made a fatal and sustained mistake.

It appeared to be a failure of the plane's pitot (pronounced pea-toe) tubes – small, forward facing ducts that use airflow to measure airspeed. On entering the storm these had apparently frozen over, blanking airspeed indicators and causing the autopilot to disengage. From then on the crew failed to maintain sufficient speed, resulting in a stall which, over almost four minutes, sent 228 people plummeting to their deaths. A

few moments later the outside air temperature plummeted, the pitot tubes iced up and an alarm sounded briefly to warn that the autopilot had disengaged.

Airbus Announces A Response To The Pilot-Automation Interface Problems; It Is a Reactive Solution; Maybe There's A Proactive One (Source: SANDY MURDOCK JUNE 26, 2014)

In the aftermath of the Air France Airbus 330 airliner crash in to the Atlantic, questions are once again being raised about the safety of flying by onboard computer systems.

Aviation safety expert and journalist David Learmount of Flight Global argues that the debate in the aviation industry is not about the safety of flying by wire, but more about how much control should stay with the pilot, at what point the computers should intervene, and what the manual control interface should be like?

Both the National Transportation Safety Board and the Federal Aviation Administration have opined that the pilot-cockpit interface has its problems due to the crewmembers' inordinate reliance on automation. Academic research has added clinical evidence that the interface needs to draw the captain and the second-in-command more into the thinking aspect of controlling the airplane.

Airbus has found 2/3 of the accidents/incidents studied) that the pilots' flying skills are deteriorating due to reliance on the automation and when they have to take full control of the plane they are not ready. Need for pilot training – an issue

recognized by industry experts. Good first step, Airbus, but there may be more that can be done to prevent the break in the pilot-automation interface.

The pilot - aircraft intelligent interface concept (Vladimír Řeřucha, Zdeněk, Dept. of Technical Cybernetics and Military Robotics, University of Defence, Brno, Czech Republic, ProceedingACMOS'05, Proceedings of the 7th WSEAS international conference on Automatic control, modeling and simulation)

The paper deals with the topical problem of increasing the effectiveness of weapon systems by constructing and providing the sophisticated man-machine interface. The man is still principal and the most important part of the man-machine system and the effectiveness of such system is first and foremost induced by the capabilities of human operator. To exert the human aptitude the appropriate intelligent man-machine interface (IMMI) is used. The main part of IMMI is the situation assessor and recognizer (SAR) providing actual information about state of man, machine and environment in form of values of situation variables. The SAR is the mapping from set of measured variables to the situation variable set and concurrently it is the model of man, machine and environment behavior. The SAR is constructed as a knowledge-based system using approaches and methods typical for Artificial Intelligence or for Intelligent Control as a branch of Cybernetics.

Pilots Rely Too Much on Automation, Panel Says - Many Aviators Have Difficulty Manually Flying Planes, Study Commissioned by FAA Finds (ANDY PASZTOR, Nov. 17, 2013)

The F-35 - The Air Force Research Laboratory's Human Effectiveness Directorate conducted tests and analysis for the F-35 Lightning II to be fitted with a speech recognition system able to "hear" a pilot's spoken commands to manage various aircraft subsystems, such as communications and navigation. Researchers at the directorate's Warfighter Interface Division collected data and recommended improvements to ensure optimal performance when the F-35's new speech recognition system undergoes future operational tests. Currently pilots must press buttons, flip switches or glance at instruments for status information. The new system not only simplifies a pilot's workload but also increases safety and efficiency, since pilots can remain focused on flying the aircraft and scrutinizing the combat environment. The speech recognition system is integrated with the aircraft's onboard computer to access data. Communication occurs through the pilot's oxygen mask microphone with command feedback provided on the pilot's helmet-mounted display.

Better User Interface Design Could Mitigate "Automation Addiction" and Other Flying Errors, HF/E Experts Suggest, Monday, December 16, 2013

Amid news reports on the National Transportation Safety Board hearings regarding possible causes of the Asian plane crash at San Francisco International Airport in July, questions have been raised about pilots' overreliance on or failure to understand cockpit automation and even whether pilots are sufficiently trained to fly without it. Eric Geiselman and colleagues propose that user interfaces that take advantage of avionics' underlying data and logic could enable pilots to better cope with

extraordinary circumstances like the unavailability of an instrument landing system, as was the case in San Francisco. In Geiselman et al.'s October Ergonomics in Design article, "Flight Deck Automation: A Call for Context-Aware Logic to Improve Safety," the authors describe prototype designs that could mitigate errors leading to accidents and incidences such as the A330 Air France Flight 447 crash in 2009 and the airport overfly of Northwest 188 that same year.

Pilot Equipment Interface - Level busts are often the result of breakdown of the pilot-equipment interface; that is to say, the incorrect handling or interpretation of aircraft equipment by the pilot. There are usually two elements to this: The pilot makes an incorrect setting or performs an inappropriate action on the equipment; and, the error is not noticed or not corrected by other flight-crew members.

Typical Scenarios

- Altimeter Setting Procedures. The pilots set an incorrect or inappropriate pressure setting on the altimeter barometric sub-scale;

- Use of the Altitude Alerter. The pilots inadvertently select the wrong altitude or flight level on the altitude alerter;

- Use of the Autopilot. The pilot enters an incorrect target altitude on the Flight Guidance System and fails to confirm the entered target on the Primary Flight Display and/or the Navigation Display;

- The pilot inadvertently arms a selected mode or selects an incorrect mode;

- The pilots become pre-occupied with the automatic systems resulting in loss of situational awareness;

Land and See: Infrared and 3-D Vision Systems Combine to Help Pilots Avoid Crash Landings - Enhanced and synthetic vision technologies (and even a combination of the two) promise to make flying on small- and medium-size aircraft safer**,** by Larry Greenemeier | February 8, 2012**,**Courtesy of NASA Langley**.**

In 2011 alone four commercial jets crashed into terrain or an obstacle, killing 140 passengers and crew, according to avionics-maker Honeywell and aerospace research firmAscend. The accidents are known as "controlled flight into terrain." Landings could be safer if new navigation displays featuring nighttime infrared imaging and 3-D graphics that accurately portray an aircraft's surroundings become standard equipment on smaller commercial and private planes. In addition to the potential safety benefits, Gulfstream, Bombardier and other makers of small and midsize business jets are also learning that the same technology can save time and money by keeping flights on schedule even in the face of weather that would normally require runway circling or flight rerouting.

Cockpit Display Interface (CDI)

Aircraft cockpit display interface (CDI) is one of the most important human–machine interfaces for information perceiving. During the process of aircraft design, situation awareness (SA) is frequently considered to improve the design, as the CDI must

provide enough SA for the pilot to maintain the flight safety. In order to study the SA in the pilot-aircraft system, a cockpit flight simulation environment is built up, which includes a virtual instrument panel, a flight visual display and the corresponding control system. Based on the simulation environment, a human-in-the-loop experiment is designed to measure the SA by the situation awareness global assessment technique (SAGAT). Through the experiment, the SA degrees and heart rate (HR) data of the subjects are obtained, and the SA levels under different CDI designs are analyzed. The results show that analyzing the SA can serve as an objective way to evaluate the design of CDI, which could be proved from the consistent HR data. With this method, evaluations of the CDI design are performed in the experimental flight simulation environment, and optimizations could be guided through the analysis.

Aircraft Pilot and Operator Interfaces, System Engineering, Avionics System Integration, Mary L. Cummings, Greg L. Zacharias, Published Online: 15 DEC 2010

The shift from mechanical/pneumatic instruments and dedicated mechanical controls/linkages to glass cockpit displays and “fly-by-wire” controls has not only transformed the appearance of the modern-day cockpit, but with the growth of on-board computational power, the role of the pilot has changed from one of directly controlling the aircraft to one of system management and automation supervision. This move towards human supervisory control means that computer-mediated

displays (primarily visual and auditory) and associated controls are the conduit through which pilots “manage” the aircraft and its subsystems. Thus operator interface design is critical for safe and efficient operation. This chapter provides a brief history of human-cockpit interface evolution, outlines design variables, principles, and guidelines for effective interface design, and concludes with a discussion of the parallels and differences between conventional manned aircraft interfaces and unmanned aerial vehicle operator ground control stations.

CHAPTER 2

BACKGROUND

2.1 Pilot's Associate Program

The Pilot's Associate program is a joint effort of the Defense Advanced Research Projects Agency and the US Air Force, managed by the Air Force's Wright Laboratory. The program began in February 1986 as an application demonstration for DARPA's Strategic Computing Initiative. DARPA wanted to advance the program's technology base, principally in the area of real-time, cooperating knowledge-based systems. The Air Force wanted to explore the potential of intelligent systems applications to improve the effectiveness and survivability of post- 1995 fighter aircraft.

Based on Dr. Bank's paper, the Pilot's Associate concept developed as a set of cooperating, knowledge-based subsystems: two assessors and two planning subsystems, and a pilot interface. The two assessors, Situation Assessment and System Status, determine the state of the outside world and the aircraft systems, respectively. The two planners, Tactics Planner and Mission Planner, react to the dynamic environment by responding to immediate threats and their effects on the pre-briefed mission plan. The Pilot-Vehicle Interface subsystem provides the critical connection between the pilot and the rest of the system. The interface

ensures that the system as a whole provides the information the pilot wants, when it is needed. Unfortunately, at that point in time, health and usage monitoring systems were not available to the degree required to assess the health status of the vehicle and recommend appropriate mitigation measures in the presence of a contingency. More recent studies are addressing various human-machine interface architectures to allocate appropriate functions between the human and the machine to reduce the effect of uncertainty. There is no published information available as to how, in the event of incipient failure conditions, automated advisories and the pilot's evidence/experience can be combined effectively and efficiently to avoid catastrophic events.

2.2 Human-machine Interface

Human-machine interface is widely used in medical and building up expert aiding systems. For instance, in S. Rerbal (2013) the author present the development a man machine telemedical interface of information and communication telemedical HMI-ICTM, which is a successful implementation for the human-machine interface used on the medical area.

Nowadays experts are providing various human-machine interface architectures to allocate appropriate function between the human and the machine to reduce the effect of uncertainty. In the paper C. Miller (2012), the authors created a prototype

demonstration to illustrate an unmanned aerial systems (UAS) control approach. This paper described the key hardware and software components, which is represented in Figure 1. This structure allows flexible switches between the described four modes, from high level to low level control. As described, multi inputs can be used, for example voice input, control bar, touch screen. For the paper G. Calhoun (2012), it is the follow up of the paper C. Miller (2012). In this paper the authors shown three evaluation cases to illustrate the approach can fulfill the ability of flexibly transition between control modes. Also introduced are topics on how to further develop the operator–automation control scheme for future UAS applications. Similar to the former paper's contribution, it demonstrates flexible switches between the four control modes. Also, this paper is intended to be applied on multi-UAS control. But based on the feedback from the UAS operators, there is a need for combine the speech recognition input with touch/gesture input to communicate the operator's intent to the automation, which is not fulfilled in the paper. Also, the voice input takes much longer time than touch screen input because of the imprecise lies in the voice input routine and natural inaccurate lies in the actions the system can get from the voice command. So when we need a better method than voice command. In our thesis we use the touch screen and built in the action set library to improve the precision and reduce the rate of mistakenly input.

Also need to be mentioned is the paper K. Sullivan (2013). In this paper the author described a method to model a parameterization of flight deck automation known as HART and link it to HAI consequences using a back propagation neural network

approach. Also, the author uses a decision tree analysis to verify the results. This approach provides a good methodology to assess the interaction and exhibits pretty good performance. Also, it can be expected to be implemented on a large number of domains where safety critical automation systems are fielded and require certification. The reason we did not apply the similar methodology is that Neural Networks are not capable of large-scale extrapolation because it requires significant investment in time and energy to fully train a neural network.

2.3 Automated System for health monitoring

Automated system could aid the pilot or the operator in many ways as they have the ability of collecting big data from numerous sensors and calculating. In the paper A. Walsdorf (1999), the author highlights the situation that the cockpits often use inappropriate automation concepts and ignore human centered design. It develops a cognitive system, which is called Crew Assistant Military Aircraft (CAMA), with human-like capabilities as the interpretation and diagnosis of the situation, planning and decision making to overcome the deficiencies in crew-machine interaction. It comprises several function units. The communication between CAMA and the crew and vice versa is controlled by another specialized module, which is described in F. O. Flemisch (1998). Several human-machine interface devices provide support for this task.

Also in the paper C. Chen (2012), the author gives an integrated architecture for the fault detection of the system. He presents a framework as the integrating software platform linking all constituent modules of the fault diagnosis and failure prognosis architecture. The inherent characteristics of the framework provide the proposed system with a generic architecture for fault diagnosis and failure prognosis for a variety of applications. With the use of Bayesian estimation theory, a generic particle-filtering-based framework is integrated in the system for fault diagnosis and failure prognosis. This is the work we borrow from the past projects from our own library. We use the similar idea and build a routine for prognosis and RUL prediction based evaluation routine. Instead of the commonly applied fault detection based methods, our method will predict the future condition, and based on the prediction, the automated system will provide estimation on the health status of the system.

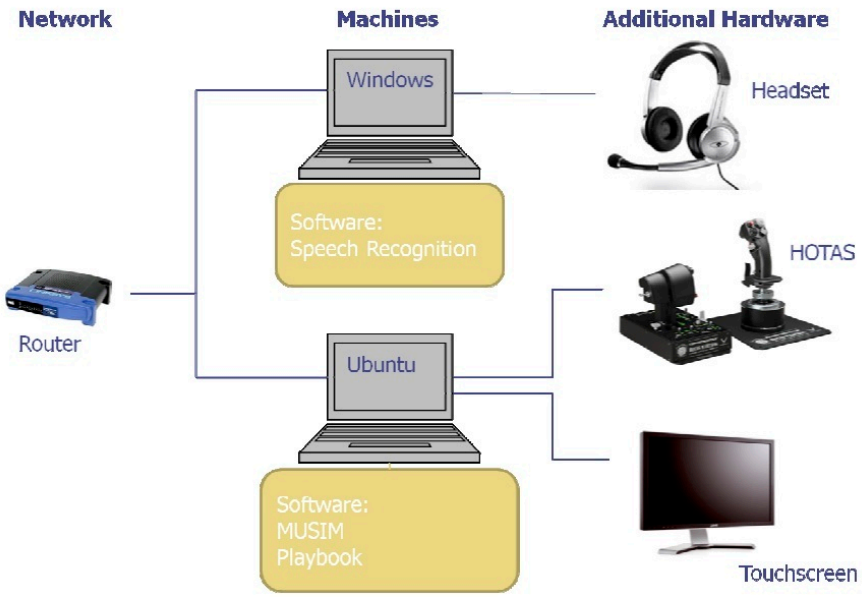
Also in B. Tristem (2010), the author presents a structure of the human-computer interaction. In this paper, the Aircraft represent two-way interactive, safety-critical, real-time systems. There is a loop of information passing as follows:

- The pilot provides an input, which the computer must interpret and use to control the mechanics of the aircraft.

- Mechanical sensors return information to the computer. This is processed and passed to the pilot for interpretation via HCI.

- Using the information presented to him by the computer, the pilot provides more input at which point the loop repeats.

The main shortcoming of this structure is that in this structure, the system will provide the reasoning result directly to the pilot, without any information fusion or combination. Thus the pilot will be the one who has to accomplish the conflict resolution when there is a conflict between the pilot's assessment and the automated system's suggestion. This is the additional information processing and computational burden for the pilot. Thus our structure will resolve the conflict in the knowledge base and provide the combined information directly to the pilot. Then the pilot can take the action without additional consideration.



- Network
  - Router: Cisco-Linksys Model BEFSR81 Cable/DSL Router with 8-PT Switch
- Machines
  - Windows Laptop:
    - MPC TransPort T2100
    - Microsoft Windows XP Professional, service pack 3
    - Intel Pentium M Processor, 1.60 GHz
    - 992 MB RAM
    - DynaSpeak v1.5.07 speech recognition software
  - Ubuntu Laptop:
    - Dell Precision M6500
    - Ubuntu 10.04 LTS
    - Intel Core i7 CPU @ 2.00 GHz
    - 3.4 GB RAM
- Additional Hardware
  - Headset: Plantronics GAMECOM Pro1
  - HOTAS: ThrustMaster WARTHOG replica USAF A-10C Throttle & Stick
  - Touchscreen[§§]:
    - 3M Touch Systems, Inc. Model M2256PW
    - 22-inch,1680 X 1050 resolution

Figure 2. Integrated Architecture for the Fault Detection of the System

2.4 Pilot's Intended Action Estimation

In the paper C.A. Miller (1999), the author cites the essentiality of monitoring world states and crew actions to keep the automated system "in the loop". Also in this paper the description of Crew Intent Estimator is given. It tracks crew behavior to infer their intent. The "Crew Coordination and Task Awareness" display consists of four small LED buttons located in the upper right portion of each pilot's main instrument panel. It reports, in text, the current inferred (1) high-level mission context, (2) highest priority pilot task, (3) highest priority copilot task, and (4) highest priority CDAS task. Pressing these buttons permits the pilot to override CIM's current inferred tasks and assert new ones via push button input. This paper did not give any graphical design of the panel or any demo on this developed schematic.

Also in the paper B. Piuzzi (2014), the author is discussing how to achieve an adequate level of human-machine cooperation to reduce new types of human errors or incidents introduced by the complex interplay of humans and automation. This is achieved by developing new concepts that balance operator workload based on the operator actual cognitive state and the environment. The structure is fully designed and tested with a demo. The panel is shown in Figure 3.

In the structures described above, we notice there is a trend to let the pilot put in tons of information into this system. However, this will introduce additional workload into this system. Moreover, it is not easy for the pilot to precisely estimate the parameter values corresponding to aircraft condition in a comparably short time. But

precision level of the information putted in will severely affect the performance of the system. This is the most critical design flaws lie in this structure. So we develop a structure that allow the pilot to input as few information as possible so that the pilot can save much more time and leave the rest of work to the knowledge base case library. The knowledge base case library built up can be done off-line.

In H. Wei (2012), a cockpit flight simulation environment is built up, which includes a virtual instrument panel, a flight visual display and the corresponding control system. Based on the simulation environment, a human-in-the-loop experiment is designed to measure the situation awareness (SA) by the situation awareness global assessment technique (SAGAT). Through the experiment, the SA degrees and heart rate (HR) data of the subjects are obtained, and the SA levels under different cockpit display interface CDI designs are analyzed. The results show that analyzing the SA can serve as an objective way to evaluate the design of CDI, which could be proved from the consistent HR data. With this method, evaluations of the CDI design are performed in the experimental flight simulation environment, and optimizations could be guided through the analysis. But based on what is demonstrated in the paper, the SA module will keep asking the pilot different questions and to figure out the system's situation. However, when the condition is severe, the pilot cannot do such large amount of reading and choose the appropriate answer. Thus it is important to help the pilot to choose from what he's familiar with and can generate the action set pretty rapidly. So our structure will allow the pilot to act faster and easily put in his action because the action set library is already settled.

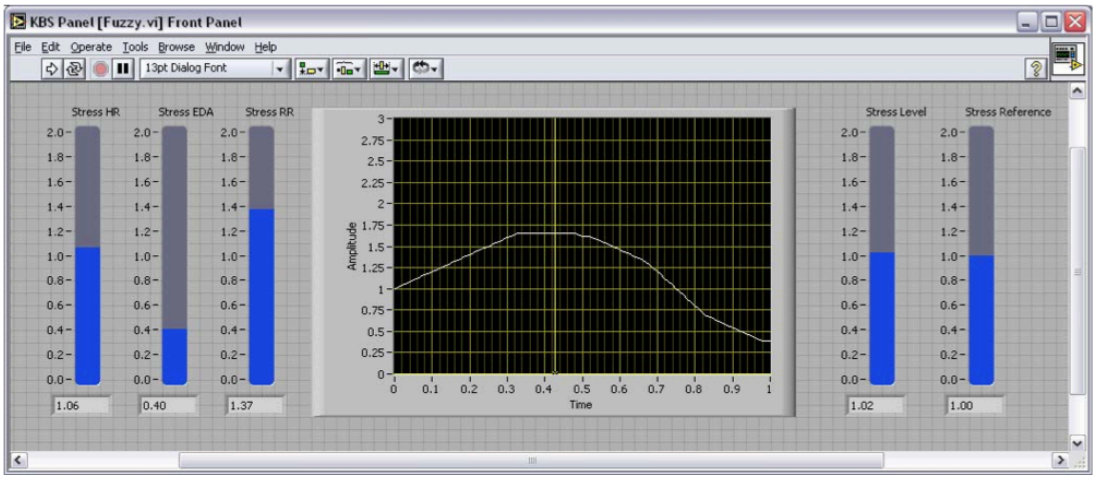


Figure 3. Knowledge Based System Panel

2.5 Information Fusions and Conflict Resolution

Most of the research work conducted in the area of decision fusion is built around Bayes Theory. Examples can be found in A.Dromigny (1997), M.A. Rodrigues (2000) and P.Lucas (2001). The Bayesian Theory is used to combine information collected from different conditions or from completely different information sources. Dempster-Shafer Theory is an extension to Bayesian theory. It is widely used in image processing, fault diagnosis and other reasoning area. In B. Yang (2006), the author presents a demo of Dempster-Shafer applied in induction motors fault diagnosis. The features are extracted from stator current and vibration. For each side they apply the neural network for diagnosis. Also, the author shows the proposed system is efficiency and has potential for real-time. But we should notice Dempster-Shafer Theory could only be applied on the combination of two information sources both on the evidence level or action level. For instance, player 1 provides an estimation of failure of each component and player 2 provides similar

estimation of these components. Then the Dempster-Shafer Theory can combine these two mass functions and come up with the combined mass. However, when the pilot only input information on the action level but the automated system can provides much more information, not only the suggested action but also evidences support it, the current approach does not support this form of information fusion/conflict resolution. So we develop a novel conflict resolution to fit our case, which is based on the Game Theory structure, to handle this problem. The Game Theory Structure is described in the following chapters.

2.6 Mass function and Dempster's rule of Combination

The mass function is the foundation of applying the Dempster-Shafer Theory on the probability analysis. But how to get the most reasonable mass function based on the data is still a not fully solved problem and yet have no general answer. Currently there are quite a lot of publications on building the mass function. In the publication of I. Bloch (1996) she cited there are generally three different levels deriving it in image processing area. The highest and most abstract level, the mass functions could be assigned by experts. At an intermediate level, mass functions are computed from attributes. This is adapted to model-based pattern recognition. At the pixel level the most widely used approach is computing masses on simple hypotheses from probabilities or from the distance to a class center, which is explained in detail in S.L Hegarat-Mascle (1997). However, this method is only limited to pure hypothesis. The

author assigned the feature vector to pure hypothesis, for instance, A is correct. Thus in the mass table, there only be values on the pure mass, which result in the belief function equals to plausibility value. In this case the author did not include uncertainty in this case. So we made an improvement here. We assign feature vectors to the other combined hypotheses, for instance, A or B is correct. Thus we can make the mass evaluation much more precise by introducing uncertainty in this structure.

There are also some other methods cited in O. Basir (2007) publication on the Engine fault diagnosis. Basically he gives an idea that estimates the mass functions by calculating the distances between the sensors values and the fault symptoms matrix. It is a reasonable methodology for quantifying the similarity between two states. However, when the state is far from any of the existing states, it is possible that the mass value is just evenly distributed, which is not reasonable. We will cite a more reasonable algorithm in the later sections.

Also in the M. Bauer (1996) the author cites that the computational complexity of reasoning within the Dempster-Shafer theory of evidence is one of the major points of criticism this formalism has to face. In order to overcome this difficulty approximation algorithms have been given aiming at reducing the number of focal elements. There are three algorithms mentioned in the paper: the Bayesian Approximation, the k-l-x method and the D1 Approximation. This could be our final step when applying our own algorithms to a real test bed.

CHAPTER 3

HMI STRUCTURE

3.1 The Human-Automation Interface

The constituent modules of the human-machine interface architecture to be pursued in the proposed effort include an on-board automated system that provides to the human operator the most accurate and reliable information regarding the platform's current and future health state through key performance metrics specific to the vehicle and onboard sensors. These are presented to the operator in a prioritized manner based on mission essential elements. A modified Dempster-Shafer formula is employed to combine conflicting and incomplete information.

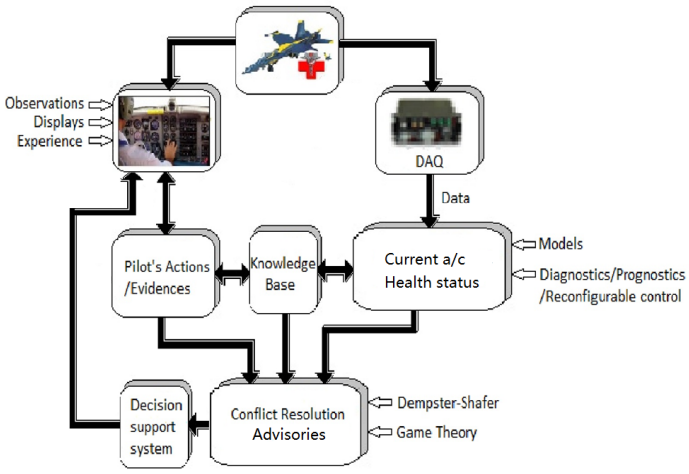


Figure 4. General Human-Machine Interface Architecture

The proposed human-machine interface architecture is illustrated in Figure 4. In the top middle of the figure is the aircraft, the targeted test bed. The pilot or operator is shown on the left. The block under the pilot represents the estimation of current system status. The latter is aided by the knowledge base which, in return, provides an

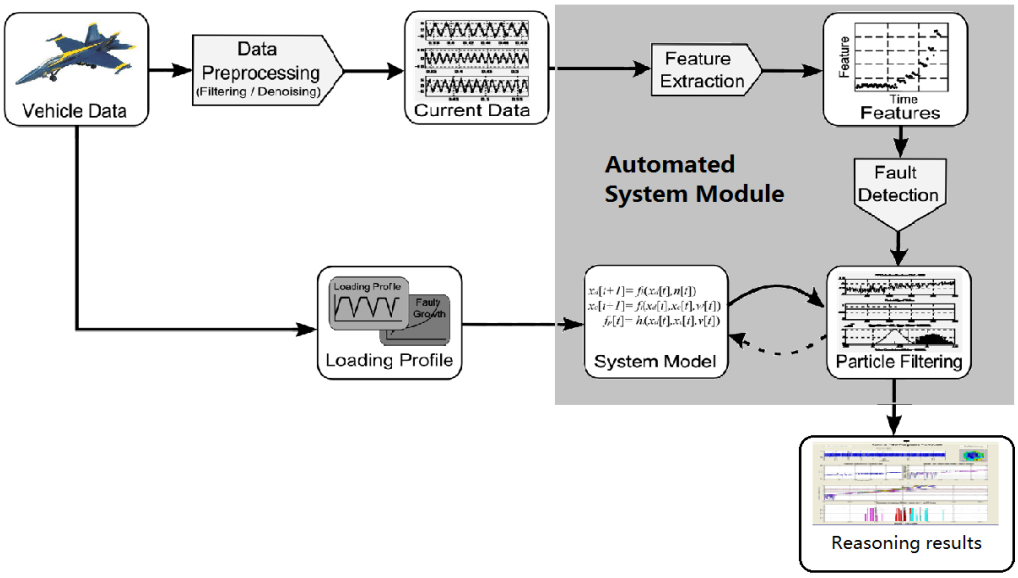


Figure 5. Structure of Automated System

input to the pilot for emergency actions. Similarly, the Data Acquisition (DAQ) module and aircraft health status estimation block are depicted on the right. There are two major information flows, i.e. information collected by the pilot and the automated system, respectively. The pilot observes current environmental conditions, reads the on-board displays, and communicates with the knowledge base. The Automated System (AS), on the other hand, gathers information from the available on-board sensor suite, represented by the DAQ module in Figure 4. The pilot and the Automated System apply then reasoning strategies based on the information collected

and data/information available in the knowledge base. If there is a conflict between the pilot's decision and the AS's advisory, the decision support system attempts to resolve such conflicts using tools from Dempster-Shafer Theory, probabilistic/fuzzy reasoning paradigms. The final recommendation is generated by the Decision Support System and sent back to the pilot as the final "decision maker" for the "best" action to mitigate the current emergency condition.

3.2 The Automated System

Figure 5 depicts the automated system reasoning modules. The goal is an advanced integrated reasoning toolset that incorporates justified levels of automated fault accommodation based on prognostic information for enhanced vehicle safety and decision support. Health and Usage Monitoring Systems (HUMS) acquire on-line in real-time appropriate data and to develop models, algorithms and software that can

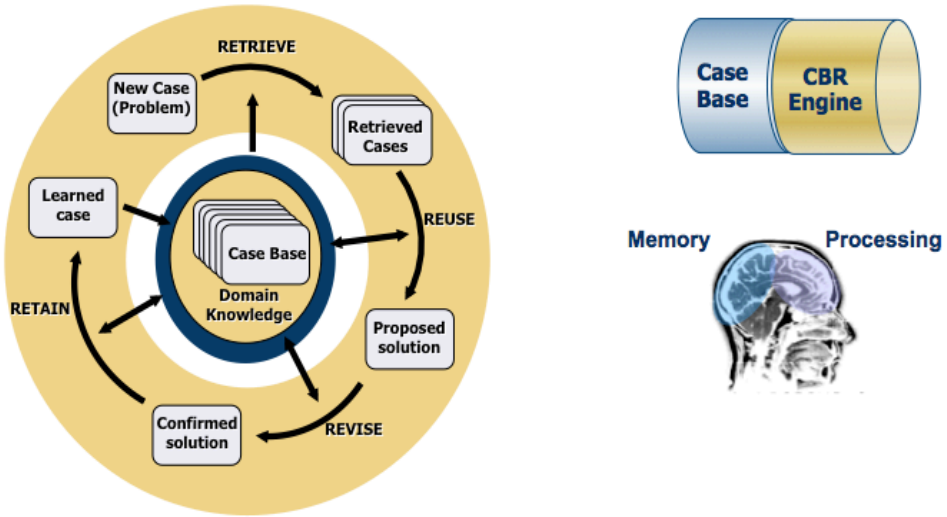


Figure 6. Schematic for Dynamic Case Based Reasoning

efficiently and effectively detect faults and predict the Remaining Useful Life (RUL) of failing components with confidence while minimizing false alarm rates. Although the pilot/operator is tasked to use his/her experience, observations and displays to decide on probable causes of an emergency condition and take appropriate initial action, the automated system must perform a series of computationally intensive processes in order to arrive at an advisory for the human operator as to the cause of current adverse conditions and appropriate mitigating strategies.

3.3 The "smart" Knowledge Base

A reasoning paradigm called Dynamic Case Based Reasoning (DCBR) that stores cases, matches new cases with stored ones and exhibits attributes of learning and adaptation will be used as the "smart" knowledge base to support learning and adaptation while providing the human operator the ability to interpret automated

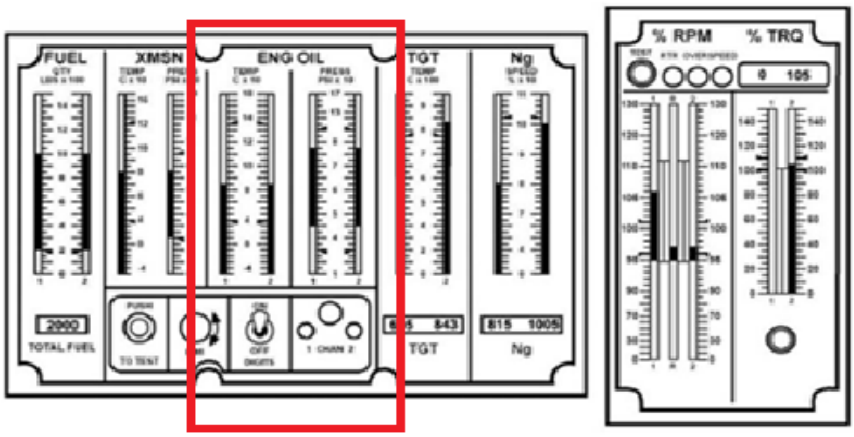


Figure 7. Displays for the pilot

system outputs correctly and to effectively control the decision making process [34]. We view this dynamic and generic knowledgebase and associated exploitation and control tools as an essential, novel, and effective way to link and exploit the human-machine information sources maximally while it serves as the "smart" strategy for accurate and robust failure detection, prediction and fault-tolerance. We pioneered the development and implementation of DCBR in fault detection and isolation of critical aircraft components. We will adapt these tools to respond to the case at hand.

3.4 The Pilot/Operator

The pilot/operator, on the other hand, gathers information in a very different way. He/she can exploit a variety of data/information sources, such as displays, alarms - red lights, personal sensing capabilities- the pilot could sense vibrations, temperature

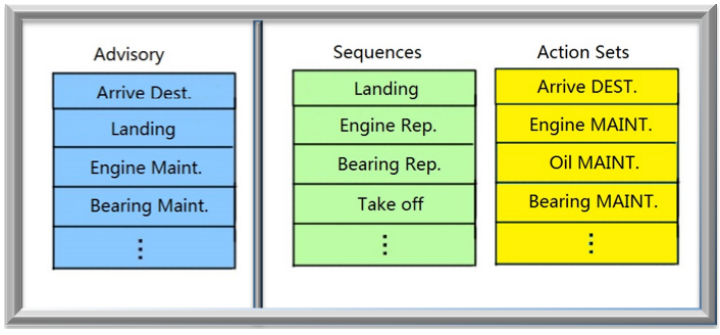


Figure 8. Control Panel for the Pilot's Action Input

rising, noise, etc., visual observations – look outside the window- rain/ snow, thunder, etc., experience, communication with ground or other aircraft. Figure 7 is an example of a typical display depicting the general form of the displayed information but

avoiding details. The pilot gathers information such as oil temperature, fuel pressure, etc. He/she uses this information to assess the current state of the system's health status and to take "initial" actions in the event of an emergency. The operator at this stage may initiate a corrective action or communicate his/her intended actions to the knowledge base. It is understood that timing requirements and sequencing of events in near real-time on-platform are crucial in the final decision making process. It is envisioned that the computational requirements burdening the AS will be minimized thus allowing for the expedient assessment of the vehicle's state and the application of conflict resolution results. The section we are developing is shown in Figure 8. On the left is the Advisory provided by the Decision support System. It is a display screen from where the pilot can read the "best" advisory, but cannot make any change. On the right is the pilot operating area. It is a touch screen and the pilot can drag needed actions from "Action Sets" area to the "Sequences" area. From the action sequences assigned in the green section, the knowledge base will know what is the pilot's action. The performance of pilot's display generated in MATLAB is shown in Figure 9.

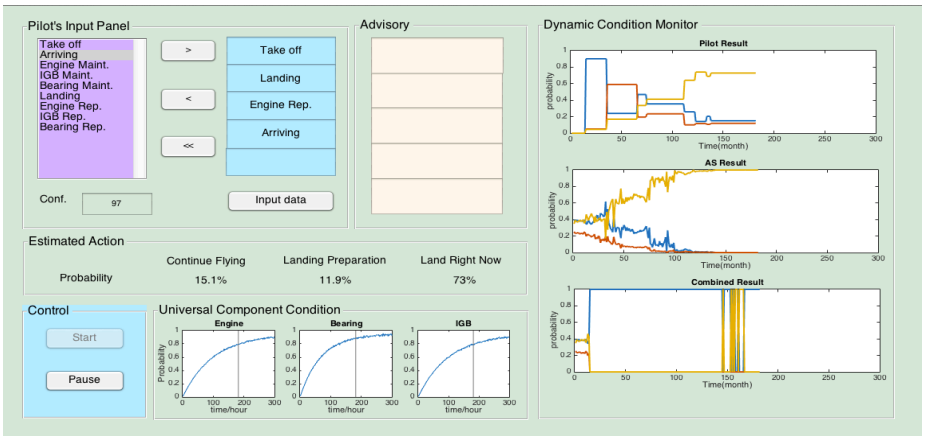


Figure 9. Graphical User Interface for the pilot

CHAPTER 4

THE AUTOMATED SYSTEM

4.1 The Health Management Module

Figure 10 depicts the automated system reasoning modules. The goal is an advanced integrated reasoning toolset that incorporates justified levels of automated fault accommodation based on prognostic information for enhanced vehicle safety and decision support. Health and Usage Monitoring Systems (HUMS) acquire on-line in

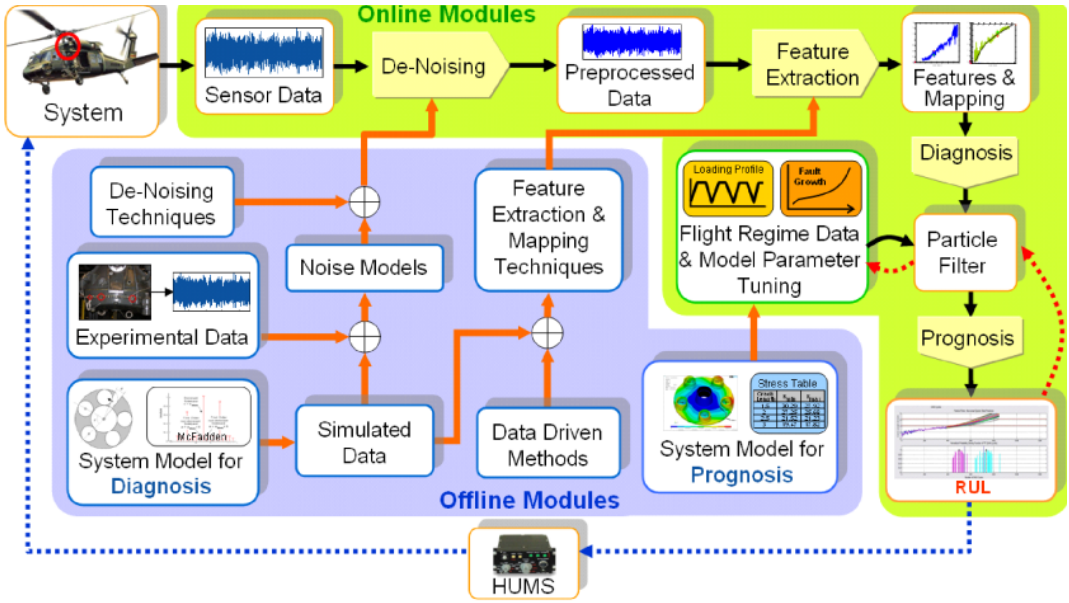


Figure 10. An Integrated PHM Architecture for Fault Diagnosis and Failure Prognosis

real-time appropriate data and to develop models, algorithms and software that can efficiently and effectively detect faults and predict the Remaining Useful Life (RUL)

of failing components with confidence while minimizing false alarm rates. Although the pilot/operator is tasked to use his/her experience, observations and displays to decide on probable causes of an emergency condition and take appropriate initial action, the automated system must perform a series of computationally intensive processes in order to arrive at an advisory for the human operator as to the cause of current adverse conditions and appropriate mitigating strategies.

4.2 An Integrating CBM+/PHM End-to-End Architecture for Fault Diagnosis and Failure Prognosis

We will introduce for purposes of this program a rigorous and verifiable framework for diagnosis and prognosis, developed, tested and applied to various laboratory, military and commercial systems at Georgia Tech that builds upon a **systems engineering process** as the driving philosophy for health management. The online modules perform raw data pre-processing, feature extraction, fault diagnosis and failure prognosis that exploit available ground truth fault data, noise models, experimental data, system models and other tools offline to tune and adapt online parameters and estimate suitable mappings [1].

4.3 Physics of Failure Mechanisms

The foundation for the development and application of PHM technologies is a

thorough understanding of the physics of failure mechanisms as critical systems are subjected to stress conditions. From the physical components/systems themselves to a good understanding of how such systems fail and under what conditions leads to optimum Condition Indicator (CI) extraction and selection and, eventually, to accurate diagnostics and prognostics.

4.4 Failure Modes and Effects Criticality Analysis (FMECA)

The starting point for "good" diagnostics / prognostics is a thorough FMECA. It describes the failure modes, sensor suite, condition indicators, possible diagnostics and prognostic algorithms. It forms the first essential step in the systems engineering process for health management of critical aircraft components/systems.

4.5 Sensors and Sensing Strategies

Sensors and sensing strategies constitute the essential requirements for fault diagnosis and failure prognosis of failing components/systems. The type, location and characteristic properties of PHM sensors, i.e. sensors that are specifically designed to monitor fault signatures, present major challenges to the system designer. We will introduce an approach to determine the type, number and location of sensing modalities that maximize the fault signal to noise ratio. We will practice these tools in our experimental studies.

4.6 Data Pre-Processing

Raw sensor data (current, voltage, vibration, temperature, etc.) must be pre-processed in order to reduce the data dimensionality and improve the (fault) Signal to Noise Ratio (SNR). Typical pre-processing routines include data compression and filtering, Time Synchronous Averaging (TSA) of vibration data, FFTs, among others.

4.7 Condition Indicator Extraction and Selection

Condition Indicator (CI) selection and extraction constitute the cornerstone for accurate and reliable fault diagnosis. The objective is to transform high dimensional raw data into tractable low dimensional form (information) without loss of useful information.

4.8 Particle Filtering for Fault Diagnosis and Failure Prognosis

The proposed fault diagnosis and failure prognosis framework builds upon mathematically rigorous concepts from estimation theory – an emerging and powerful methodology in Bayesian theory called Particle Filtering that is particularly useful in dealing with difficult non-linear and/or non-Gaussian problems. [18] Particle filtering facilitates the estimation of the state (fault) model over consecutive time instants as

measurements become available. The particle filtering routines for diagnosis and prognosis are implemented and executed in near real-time and constitute an integrated framework where the results of diagnosis serve as the initial conditions for prognosis in a transparent and efficient manner [15,16,17,18].

4.8.1 Fault Diagnosis

The proposed particle-filter-based diagnosis framework aims to accomplish the tasks of fault detection and identification using a reduced particle population to represent the state probability density function (pdf) [19,20,21,22]. This framework provides an estimate of the probability masses associated with each fault mode, as well as a pdf estimate for meaningful physical variables in the system. Once this information is available within the diagnostic module, it is conveniently processed to generate proper fault alarms and to inform about the statistical confidence of the detection routine. Customer specifications are translated into acceptable margins for the type I and II errors in the detection routine. The algorithm itself will indicate when the type II error (false negatives) has decreased to the desired level. Figure 11 shows the anomaly detection results based on an RMS feature. The first plot depicts the progression of the feature as a function of time while the second is the probability of failure; the last one shows the baseline and fault pdfs at 5% false alarm rate. The Type II error is 1.1117% at that specific instant of time. Another performance metric is the Fisher Discriminant Ratio shown at the bottom of the figure.

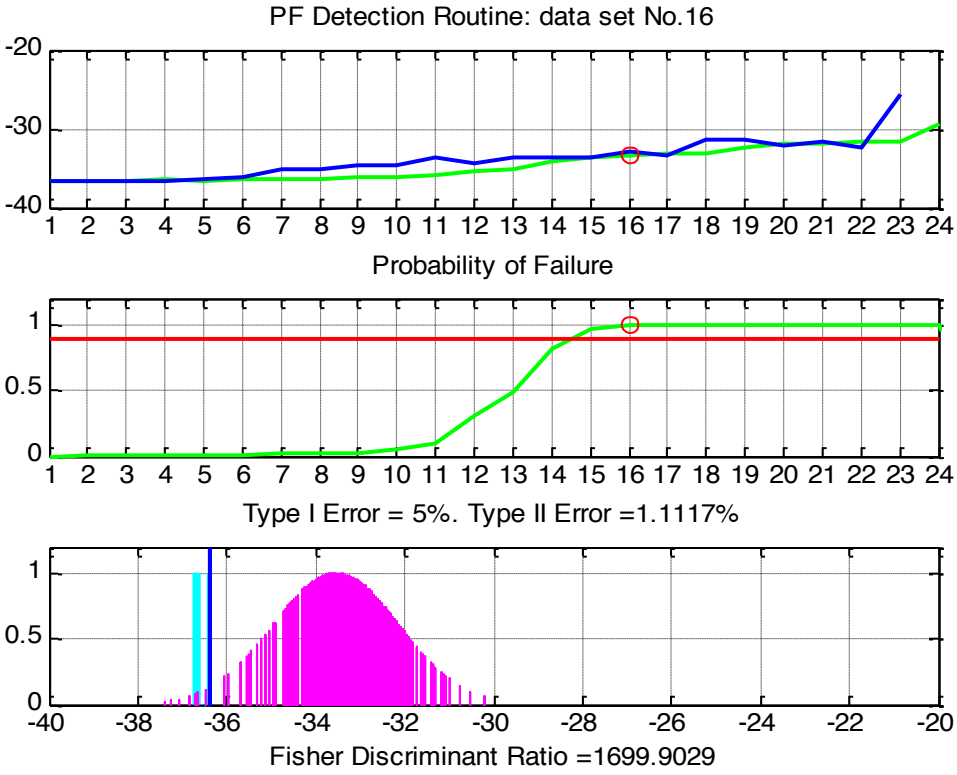


Figure 11. Typical Diagnosis Result

4.8.2 Failure Prognosis

The evolution in time of the fault dimension may be described through the state equation,

$$\begin{cases} x_1(t+1) &= x_1(t) + x_2(t) \cdot F(x(t), t, U), w_1(t) \\ x_2(t+1) &= x_2(t) + w_2(t) \end{cases}$$

where $x_1(t)$ is a state representing the fault dimension under analysis, $x_2(t)$ is a state associated with an unknown model parameter, U are external inputs to the system (load profile, etc.), $F(x(t), t, U)$is a general time-varying nonlinear function, and $w_1(t)$, $w_2(t)$are white noises (not necessarily Gaussian) [23,24]. The nonlinear function $F(x(t), t, U)$may represent a model based on first principles, a neural

network, or even a fuzzy system.

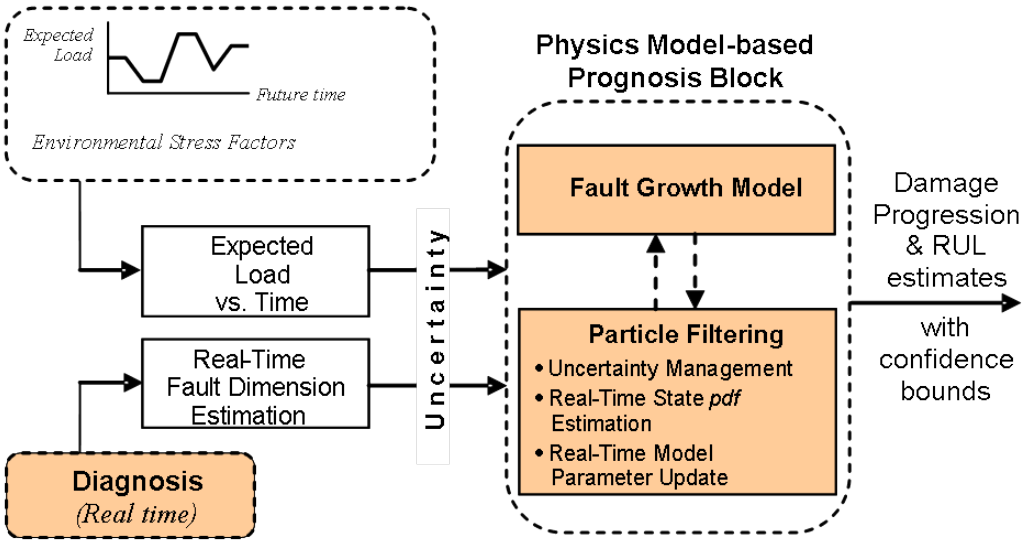


Figure 12. The Prognostic Framework

4.9 Case Study: Oil Degradation Analysis

In our example, we shall discuss the intermediate gearbox of the H-60 (Black Hawk) helicopter. The H-60 helicopter plays a pivotal role in a variety of missions for the U.S. armed forces. Variations of the helicopter include Black Hawks, Pave Hawks, and Seahawks. The manufacturer, Sikorsky Aircraft

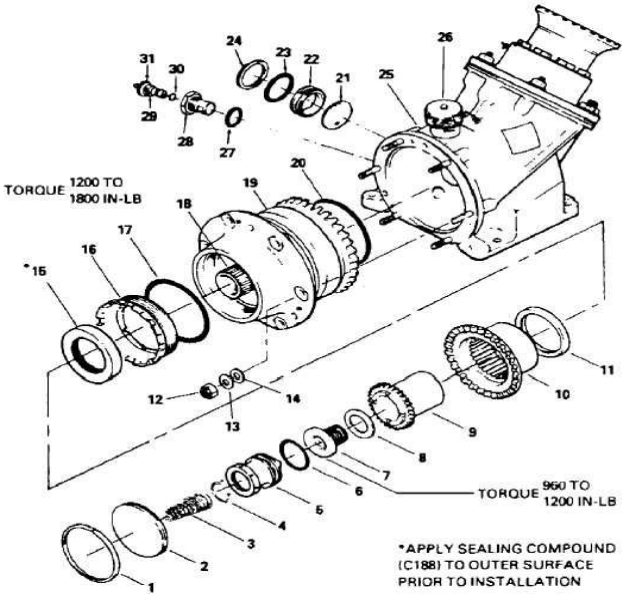
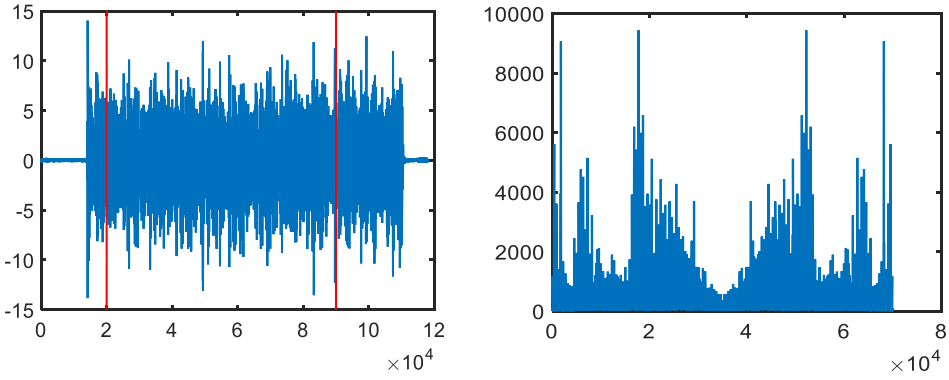


Figure 13. Schematic of UH-60 IGB

Corp., says more than 2,500 H-60s are in service with the U.S. Army, Navy, Coast Guard, Air Force, and Marine Corps. A schematic view of the H-60 IGB is shown in Figure 13.

Data Collection and Data Pre-processing

The raw data are collected in the past projects in our lab. For each data set in time domain, (e.g. in Figure 14(a)), we first segment the period with signal. For instance, the useful information is the period between the two red lines. Then we apply the Fourier Transform and come up with the plot of the frequency domain, which is shown in Figure 14(b).



(a) (b)

Figure 14. Fourier Transform of the preprocessed signal

Feature Selection and Extraction

First we pick up 10 features as the feature candidate. On the time domain we pick the

mean value and the standard deviation. One the frequency domain, we pick the amplitude and the spectrum density at 1× shaft speed, 2× shaft speed, 3× shaft speed and the location with maximum energy. Then we calculate the correlation coefficient of each feature and come up with the table attached.

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}
CC	0.7972	0.3797	0.7867	0.5902	0.2729	0.0327	0.0097	0.2957	0.2368	0.9347

We pick up the three features with highest correlation coefficient, which are f_1, f_3, f_{10}, as highlighted in yellow.

The next step is we use the feature combination method and come up with a combined feature:

$$F_{combined} = f_1^{0.29} \cdot f_3^{0.18} \cdot f_{10}^{2.01}$$

The correlation coefficient is 0.95, which is higher than any individual feature.

The Diagnostic and Prognostic Algorithms

Degradation detection and prediction require an appropriate estimation technique, in addition to data and a model, as pointed out above. Estimation methods have been developed over the past years such as Kalman filtering, regression tools, etc. to

address the prognosis problem. We take advantage in this study of a novel estimation method called particle filtering that has been shown to outperform other known methods while dealing with difficult nonlinear and/or non-Gaussian problems (3). The underlying principle of the methodology is the approximation of relevant distributions with particles (samples from the space of the unknowns) and their associated weights. This is of particular benefit in diagnosis and prognosis of complex systems, because of the nonlinear behavior when operating under fault or degradation conditions.

A fault or parameter degradation diagnosis procedure involves the tasks of degradation detection, isolation and identification (assessment of the severity of the degradation). At any given instant of time, this detection framework provides a probabilistic estimate of the fault or degradation mode. Once this information is available, it is processed to generate proper fault alarms and to inform about the statistical confidence of the detection routine. Furthermore, estimates for the system continuous-valued states (computed at the moment of degradation detection) may be used as initial conditions in prognostic routines. Customer specifications are translated into acceptable margins for the type I and/or II errors, i.e. the false alarm and confidence or accuracy of detection.

Parameter degradation detection procedure:

Use data for IGB gear crack to derive a histogram (approximation to a probability density function) for baseline conditions. Store the baseline histogram and repeat the

histogram calculation as new data is streaming in. At the beginning of detection, compare the current situation (histogram) with the baseline; set the false alarm rate at (say) 5% and declare the detection of the degrading parameter when the confidence or accuracy reaches 90%.

Prognosis of degradation evolution

Prognosis possesses the ability to predict accurately and precisely the future condition and remaining useful life of a degrading parameter. Since prognosis projects the current condition of the fault indicator in the absence of future observations and necessarily entails large-grain uncertainty, it is considered widely as the Achilles' heel of Condition Based Maintenance. Prognosis may be understood as the result of the procedure where long-term (multi-step) predictions - describing the evolution in time of a parameter degradation indicator – are generated with the purpose of estimating the Remaining Useful Life (RUL) or time to check the IGB. The same particle filtering framework and nonlinear state model suggested above will be used to estimate the RUL (4, 5). The state estimation is achieved recursively in two steps: prediction and update. The prediction step is intended to obtain the prior probability density function of the state for the next time instant.

The detailed algorithm steps for condition prognosis may be stated as:

Step 1: The Symbolic Regression model is trained with available condition

data to model the IGB crack growth process.

Step 2: The crack growth model is employed in the particle filtering formulation to draw a set of particles. According to the values of the particles and current weights, condition prediction is carried out next. When a new measurement becomes available, the weights of the particles or samples are calculated.

Step 3: Update the process noise and model parameters.

Step 4: Repeat *Step2* and *Step 3* until prognosis is complete.

2.6 Simulation Results

We describe in this section simulation results derived from the application of the tools and methods suggested previously.

Detection Results

The baseline result is generated from data set without gear crack. We use this data set to generate a histogram as the baseline distribution.

The Detection result is shown in Figure 15. Step 2, i.e. prediction, is triggered after the detection step is completed before the dashed line. Prognosis is triggered after 90 hours.

The blue histogram is the baseline distribution and the red one is the current distribution. The black line is the threshold which set the Type I error or false alarm at 5%. From the figures, it is observed that for the gear crack, at time 991h, the Type II error, or accuracy/confidence, is reduced to 8%, which implies that the confidence of a degrading state being present has increased to 92%. So at time 991 hours, the IGB gear crack abnormal is declared.

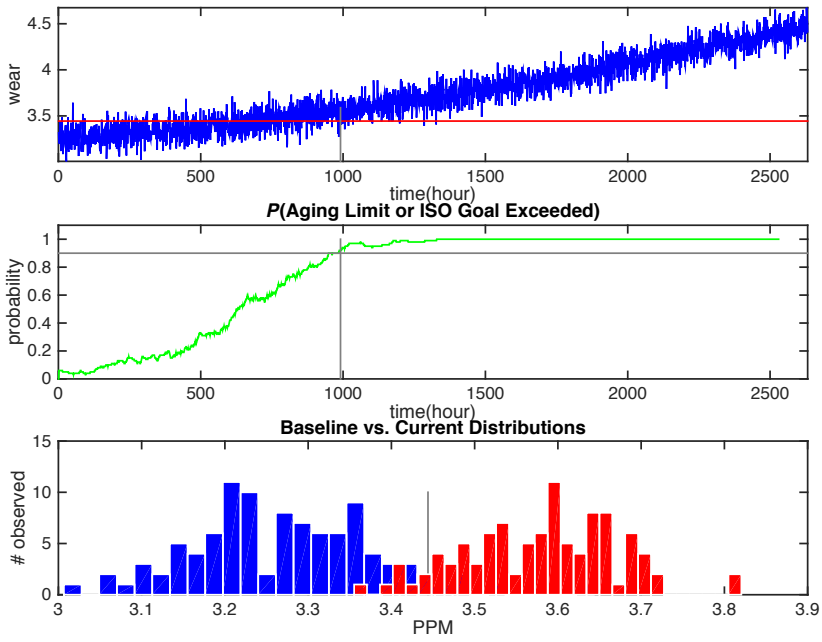


Figure 15. Diagnosis Result

Prediction results

Figure 16 is the result of model tuning and prediction for wearing. Time is divided into three sections, which are separated by dashed line at time 991h and 2637h,

respectively. Section 1 is the fault detection period we mentioned. At section 2, the model-tuning module is tuning the model parameters. The blue line and green line are upper bound and lower bound of gear wearing level, respectively. The black dots are the measured wear levels. As shown in the figure, the tuned model fits the data well. The red horizontal line is the threshold specified by the user; it designates when the evolving degradation curve, projected on the time axis, reaches the limit of "normal" IGB operation and the IGB is in a faulty condition. The prognosis horizon is shown in the following figure. We tune the model parameters to match the model prediction results with incoming data. The tuning routine employs the error between the model output and the actual data in an optimization scheme to ascertain that the model is a true representation of the current parameter behavior. Model Tuning is shown before the second dashed line. Prognosis is triggered after 2637 hours. The horizontal markers at 5 designate the hazard zone, i.e. the wear level that indicates the IGB is at faulty. It is a pdf specified by the user and based on historical data and experience.

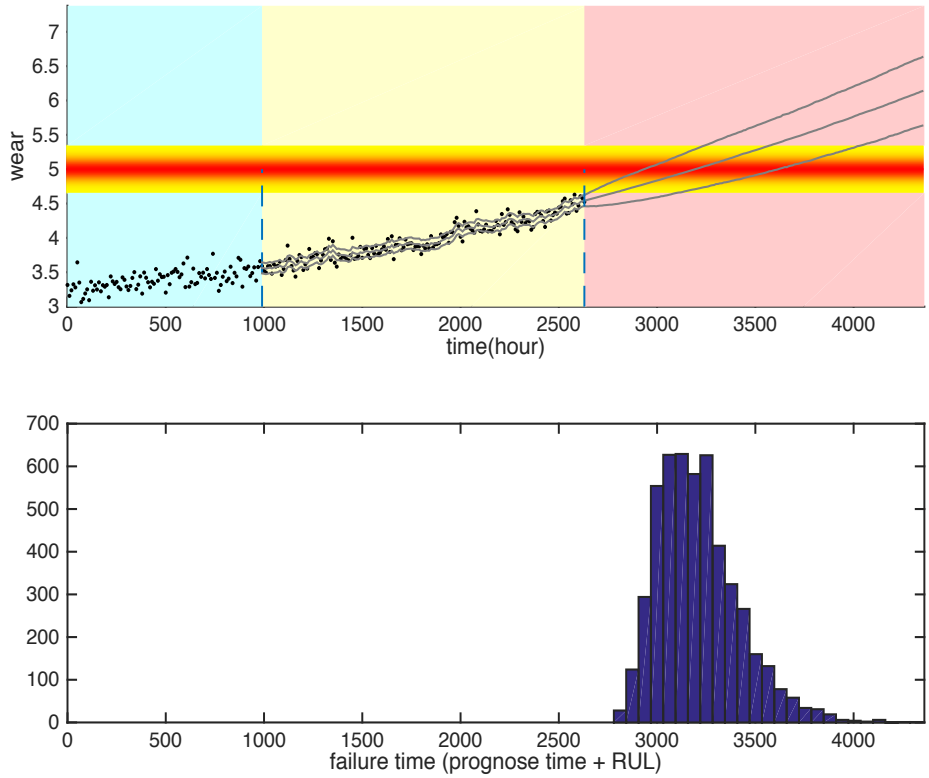


Figure 16. Prognosis Result

Computational Burden

We tested above routine in MATLAB and come up with the attached estimation of whole procedure runtime.

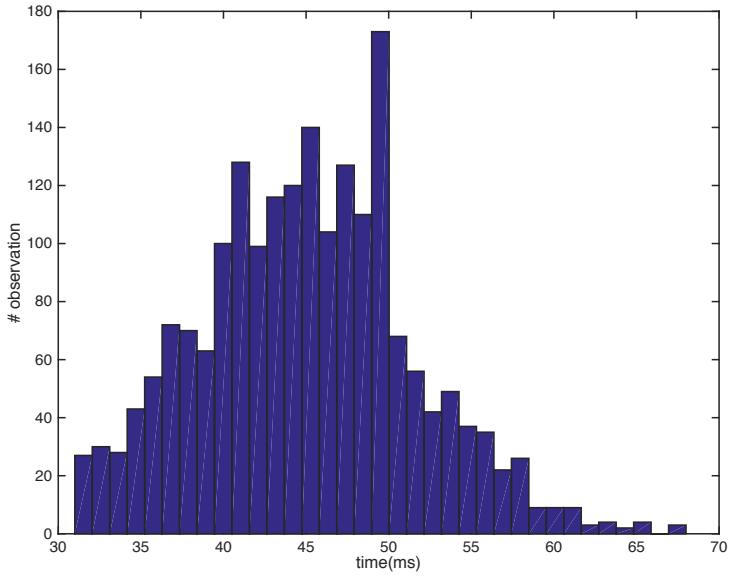


Figure 17. Computational Burden

The mean value is 45.03ms and the standard deviation is 6.43ms. So we know this procedure is on a 1×10^{-2}seconds level.

CHAPTER 5

THE PILOT'S INTERFACE

5.1 Pilot information collection

The pilot/operator, on the other hand, gathers information in a very different way. He/she can exploit a variety of data/information sources, such as:

- Displays
- Alarms -red lights
- Personal sensing capabilities- the pilot could sense vibrations, temperature raising, noise, etc.
- Visual observations – look outside the window- rain/ snow, thunder, etc.
- Experience
- Communication with ground or other aircraft.

In addition, based on the talking with an experienced pilot, we figured out there are a lot of aiding systems developed to help the pilot to be aware of the condition of the aircraft. For instance, there is a suit, when the pilot feels the temperature is rising at leg area, it indicates there's something wrong with the engine; the back area is heating up means there's something wrong in the transmission area. Also there are some goggles and headphones can display or tell the abnormal to the

pilot.

The operator at this stage may initiate a corrective action or communicate his/her intended actions to the knowledge base. It is understood that timing requirements and sequencing of events in near real-time on-platform are crucial in the final decision making process. It is envisioned that the computational requirements burdening the AS will be minimized thus allowing for the expedient assessment of the vehicle's state and the application of conflict resolution results.

The pilot's interface is shown in Figure 10. It consists of several sections. The Pilot's Input Panel is the section where the key actions are placed into the system. The operator can use the arrow buttons to add or delete actions and the sequence is generated in the blue area on the right. Also the confidence level is inserted into the table. The pilot is exposed to the confidence estimate for each one of his/her suggestions.

The interface software estimates the pilot's intended actions and displays the probabilities at the Estimated Action column. Such probabilities are displayed as Continue Flying, Landing Preparation and Land Right Now. Precision Level is set at 0.1%. Changes of the estimated action with time are displayed at the Dynamic Condition Monitor Section. Also displayed are the Automated System's advisory and the final advisory after conflict resolution, if present.

5.2 Sequence Generation

For each Action set, firstly, all possible causes are listed, accompanying the specific action (e.g. land the aircraft) chosen to be executed. The corresponding sequences are provided next. When the system is attempting to estimate the pilot's intended action, the action routine proceeds in the inverse direction. The system will collect the sequences and estimates the possible causes. The causes are used to evaluate the possible pilot's intended action, as shown in Figure 18.

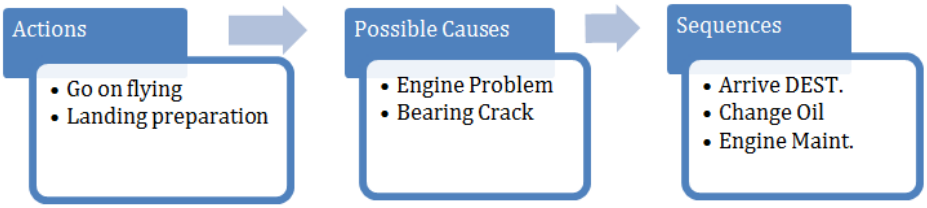


Figure 18. Sequence Generation Structure

When detailed cases are filled in, the resulting demonstration entries are shown in Table 2. This table can be used for tracking the action set resulting from the given sequences.

Table 2. Action Set and Sequences Table

Action Set	Possible cause(s)	Sequences
Go on flying	Condition is good enough	Take Off
		Arrive DEST.
		Landing
Landing Preparation	Bearing Crack to Med. Level	Take off
		Arrive DEST.
		Landing
		Bearing MAINT.
	IGB to ill condition	Take off
		Arrive DEST.
		Landing

		IGB MAINT.
Landing Right now	Bearing Crack to Severe Level	Take off
		Landing
		Bearing REP.
	Engine failure	Take off
		Landing
		Engine REP.

5.3 Coding Scheme

Using these structures, a detailed methodology for the communication between the pilot, the automated system, and the knowledge base, is suggested next. Coding is an important consideration in the development of the communication strategy facilitating the proper transmission of information. The proposed coding scheme is shown in Figure 19. All other models of the overall architecture are integrated into the central PC and present no major communication problems; only the communication between the pilot and the knowledge base requires special attention. The proposed method is illustrated via the following examples.

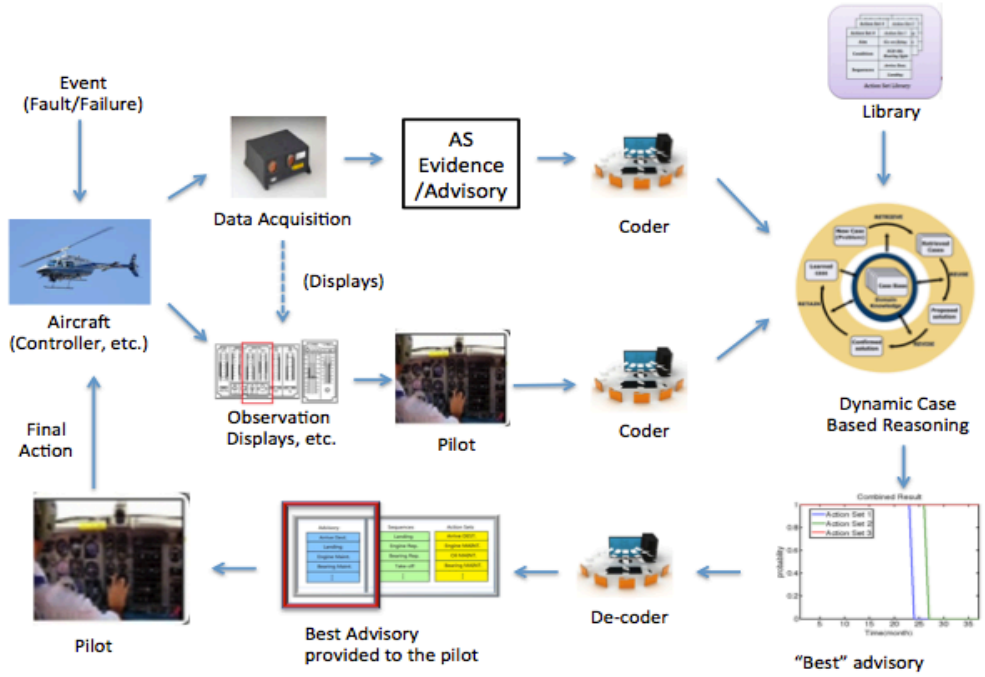


Figure 19. Coding Schematic

5.4 A demonstration of the coding scheme

The pilot enters his/her assessment of the current health state and the recommended actions via a touch screen. The recommended actions are coded in MATLAB by a set of rules, as shown in Figure 20, and depicted as a screenshot.

```
% %%%%%%%%%%Table for reference%%%%%%%%%%%%%
%              Take off      -- 1
%              Arriving      -- 2
%              Engine Maint. -- 3
%              IGB Maint.    -- 4
%              Bearing Maint.-- 5
%              Landing       -- 6
%              Engine Rep.   -- 7
%              IGB Rep.      -- 8
%              Bearing Rep.  -- 9
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Figure 20. Example of MATLAB coding

For instance, the action sequence

[Take off, Arriving, Landing]

is coded as:

[1,2,6, NaN, NaN]

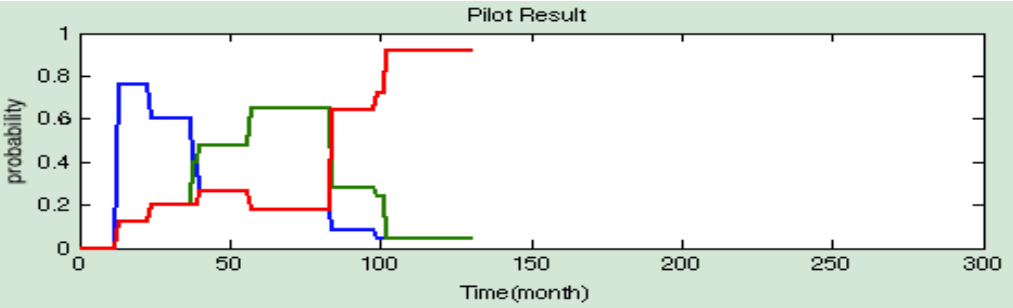


Figure 21. A demo of Estimated Pilot's Intended Action Estimation

In parallel, the pilot's confidence in his suggested actions is also sent to the central module and saved as a double format number, to be described in the sequel.

The system finally generates the probability of taking different actions at each time instant and the corresponding advisory is provided as shown in Figure 21.

CHAPTER 6

DYNAMIC CASE BASED REASONING

6.1 Cases in the Knowledge Base

A reasoning paradigm called Dynamic Case Based Reasoning (DCBR) that stores cases, matches new cases with stored ones and exhibits attributes of learning and adaptation is used as the "smart" knowledge base to provide the human operator the ability to interpret automated system outputs correctly and to effectively control the decision making process. The knowledge base contains two sub sections: Pilot's library and AS's library. Cases in the Pilot's library contain the pilot's recommended sequences, pilot's confidence and suggested action set. The AS's library contains the key sensor data and the corresponding suggested actions. The structures of the cases are shown in Table 3.

Table 3. Cases in Knowledge Base

Categories	**Sequences**					**Confidence**	**Suggested Action**		
Subitems	S_1	S_2	S_3	S_4	S_5	Conf.	p_1	p_2	p_3

Categories	**Features**					**Suggested Action**		
Subitems	f_1	f_2	f_3	f_4	f_5	p_1	p_2	p_3

6.2 Similarity Metrics

The task of the knowledge base is to discover similarities between a new case and those stored in the case library. For this purpose, appropriate similarity metrics are defined and exploited. For the automated system, distance based criteria are defined and applied. Basically, the similarity metric for the automated system cases is defined as:

$$d(f, f_c) = \sqrt{\frac{\sum_{i=1}^{N} w_i \cdot \left(\frac{f_{ci} - f_i}{f_i}\right)^2}{\sum_{i=1}^{N} w_i}}$$

Here f and f_c are two feature vectors, of N dimensions each, with w_i the weight for each dimension. Once the distance is obtained, the inverse of the distance is normalized to 1 and the mass for each case is calculated.

For the pilot's case, in order to determine the similarity of two sequences of action, the first step is to determine if the components are the same and if they are of the same order. We call these two properties as similarity of components (SC) and ratio of ascending order (RAO), respectively. For the similarity of components property, we compare the components of the two sequences and determine the number of actions that are the same. For instance, if

$$a = [Take\ off, Arriving, Landing, Engine\ Maint.],$$

$$b = [Take\ off, Landing, Arriving, IGB\ Maint., NaN]$$

then,

$$SC(a,b) = \frac{3(\#\ of\ same\ actions)}{4(length\ of\ the\ longer\ sequence)}$$

It is noted that, when the sequence has a NaN term, this action is not counted in the length of the sequence.

The next property is related to the ratio of ascending order. The pilot enters a sequence a_R into the system,

$$a_R = [Take\ off, Arriving, Landing, Engine\ Maint.]$$

which is then translated into a sequence of numbers:

$$a_R = [1,2,6,3]$$

Now, we compare each set of two numbers in the vector and count the number of ascending pairs:

$$n_{asc} = \#of\ ascending = \{(1,2),(1,6),(1,3),(2,6),(2,3)\} = 5$$

As there are four numbers in the vector, the total number of pairs is:

$$n_{total} = C_4^2 = 6$$

So, the ratio of ascending order is:

$$RAO(a_R) = \frac{n_{asc}}{n_{total}} = \frac{5}{6} = 0.87$$

An additional factor, called tuning factor, is also considered. This factor is related to

the confidence level. The expression for the tuning factor is:

$$t_{tunning} = log\frac{1}{|conf_a - conf_c|}$$

The closer $conf_a$ is to $conf_c$, the larger is $t_{tunning}$. The reason for choosing the log function instead of $\frac{1}{|conf_a - conf_c|}$ is that log function is smoother. For instance, if $conf_a$ =0.9, $conf_{c1}$= 0.91, $conf_{c2}$= 0.901, then the terms $\frac{1}{|conf_a - conf_c|}$ will be 10 and 100. But t_{tuning} will be 2 and 3, resulting in a smaller difference, a more reasonable outcome.

In summary, the similarity metric defined is shown in the following equations. The mass is the probability of case a being the same as case c, where t_{SC} is the term related to the similarity of components, t_{RAO} is the term related to the ratio of ascending order, $t_{tunning}$ is the confidence related information; ω_i is the weight for each item:

$$mass(a,c) = \begin{cases} \frac{t_{tunning}}{\omega_1 t_{sc} + \omega_2 t_{RAO}} & , \quad if \;\; t_{sc} = t_{RAO} = 0 \\ A \cdot t_{tunning} & , \quad otherwise \end{cases}$$

$$t_{tunning}(a,c) = \begin{cases} log\frac{1}{|conf_a - conf_c|}, if \;\; conf_a = conf_c \\ A \quad , \quad conf_a \neq conf_c \end{cases}$$

$$t_{sc}(a,c) = 1 - SC(a,c)$$

$$t_{RAO}(a,c) = \begin{cases} \frac{|RAO(c) - RAO(a)|}{RAO(a)}, & if\ RAO(a) \neq 0 \\ |RAO(c) - RAO(a)|, & RAO(a) = 0 \end{cases}$$

Here, A can be set as a very large positive number.

CHAPTER 7

CONFLICT RESOLUTION METHODOLOGY

7.1 The Automated System-Pilot Conflict Resolution Methodology

Conflicts arise between the pilot's intent/commands and automated system commands/advisories. They arise from the different perceptions of the pilot and the

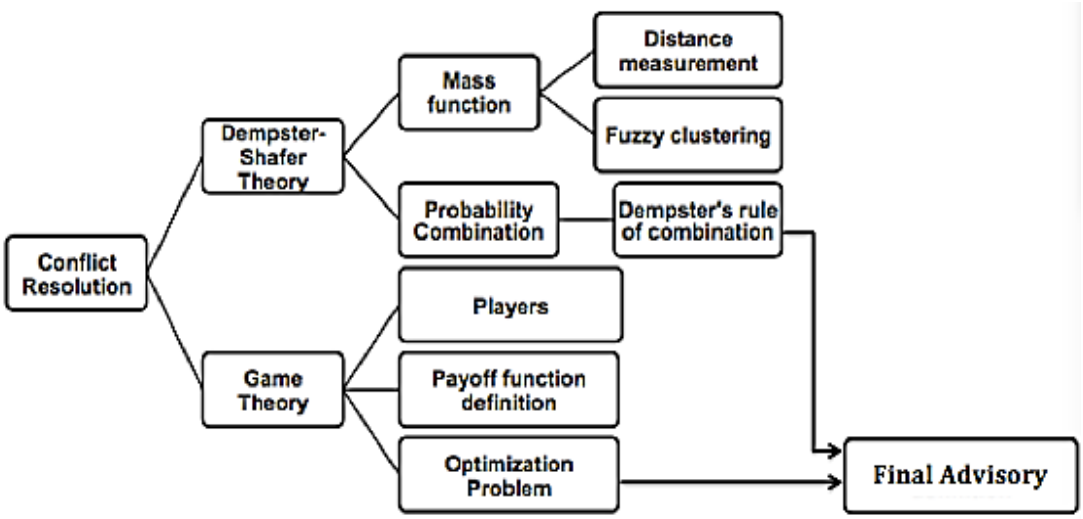


Figure 22. Conflict Resolution Structure

automated routines stemming from experience, current data and information available to the pilot and the control architecture which may differ in content, quantity and means for the expedient presentation and follow-up action. The principal task of the decision support system is, therefore, to resolve conflicts between the pilot's actions and those recommended by the automated system. Conflict resolution is a challenging task that must be addresses methodically in the presence of incomplete evidence,

ambiguity and noise. We may apply such methodologies as Dempster-Shafer Theory and Game Theory, among others.

7.2 Dempster-Shafer Theory

The Dempster-Shafer Evidential Theory is widely used in possibility combination, sensor fusion, artificial intelligence, and conflict resolution areas. It allows one to combine evidence from different sources and arrive at a degree of belief that takes into account all the available evidence [31,32,33]. In this formalism a degree of belief, which is also referred to as a mass, is represented as a belief function. Possibility values are assigned to sets of possibilities rather than single events. Dempster-Shafer theory assigns its masses to all non-empty subsets of entities. The architecture of the Dempster-Shafer algorithm is shown in Figure 22. Application of the Dempster-Shafer Theory requires first and foremost the calculation of the mass functions, as detailed in the sequel. Methodologies such as distance-based measurements can be considered to assess the mass function structure. Fuzzy notions or probabilistic design methods may be employed to categorize the fault modes, and then distance-based measurements are exploited to calculate the mass functions. Finally, with the mass function structure at hand, Dempster's rule of combination is applied to address the conflict resolution problem. We outline below the basic structure of the architecture.

7.2.1 Dempster's Rule of Combination

Assume m_1 and m_2 are two belief function structures on X provided by the pilot

and automated system, respectively. m_1 has focal elements $A_1, \cdots, A_k$ and m_2 has $B_1, \cdots, B_p$. We will introduce a modified form of Dempster's rule to combine evidences and avoid counterintuitive results faced by classical methods. Consider two mass functions m_1 and m_2 and define: $m = m_1 \perp m_2$, where $\perp$ denotes the direct sum and m is calculated as:

$$K = \sum_{\substack{A_i, B_j \\ A_i \cap B_j = \emptyset}} m_1(A_i) m_2(B_j)$$

$$m(A) = \sum_{\substack{A_i, B_j \\ A_i \cap B_j = A}} m_1(A_i) m_2(B_j), \qquad A \neq \emptyset, X$$

$$m(X) = \sum_{\substack{A_i, B_j \\ A_i \cap B_j = X}} m_1(A_i) m_2(B_j) + K$$

$$m(\emptyset) = 0$$

It can be shown that the new rule is more stable than Dempster's original rule. If there is a slight change in the belief structure of the individual masses there is only a corresponding slight change in the resulting belief structure.

7.2.2 Mass function Evaluation

The mass function is the foundation for applying Dempster-Shafer theory to the conflict resolution problem. The estimation of the mass functions is a challenging problem addressed by several investigators without a satisfactory solution from an analytical and computational perspective. The following sections detail its principal components. In O. Basir [28] the author suggests such methods for diagnosis of

engine fault modes. This contribution estimates the mass functions by calculating the distances between the sensor values and the fault symptoms.

Distance based mass function estimation

Distance based reasoning is the standard algorithm for the mass calculation. A similar algorithm is described in O.Basir [28]. We take advantage of a classification-based method that allows the use of inaccurate information and produces more accurate results than a Bayesian structure.

Let $X = [x_1, x_2, \dots, x_n]$ represent the state of the component we are monitoring; x_i is the ith feature that describes an aspect of the system state, i.e. oil cooler temperature, crack size; n is the number of features. Assume that we have M fault modes. Assign mass values to $\sum_{k=1}^{M} \binom{M}{k} = 2^M - 1$ hypotheses. We define the label vectors as:

$$L = [l_1, l_2, \dots, l_M]$$

For each l_i element in L, $l_i = 1$ means that there is a possibility of the ith fault mode happening. For example, if L= [1, 1, 0, 0, … , 0], the hypothesis is "Fault 1 or Fault 2 occurs".

The next step addresses the determination of the state vector X for each hypothesis. First, hypotheses with only one element are considered. In other words, calculate $m(A)$ for $card(A) = 1$. The label vector for this condition satisfies:

$$\|L\|_2 = 1$$

Based on the given database, the mean state vector for each fault mode is calculated. Thus, we arrive at the mean basis:

$$\{X_{1c}, X_{2c}, \cdots, X_{Mc}\}$$

Now, define the base matrix as:

$$X_B = \begin{bmatrix} X_{1c} \\ X_{2C} \\ \vdots \\ X_{MC} \end{bmatrix}$$

Consider next the hypotheses with multiple elements. For each hypothesis j we have the label vector L_j. Based on this, the state vector X_j of this hypothesis may be calculated by:

$$X_j = \frac{L_j X_B}{\|L\|_2} = \frac{[l_1, l_2, \dots, l_M] \begin{bmatrix} X_{1c} \\ X_{2C} \\ \vdots \\ X_{MC} \end{bmatrix}}{\|L\|_2} = [x_{j1}, x_{j2}, \dots, x_{jn}]$$

This process leads to the state vector for all hypotheses.

On the other hand, the measured state vector could be determined from the sensor measurements:

$$S = [s_1, s_2, \dots, s_n]$$

The more similar is S to X_j, the more probable is the jth hypothesis. The distances between all sensor measurements and all faults can be captured in a matrix form:

$$D = [d_1, d_2, \dots, d_{2^M-1}]$$

The smaller is d_j, the more probable is the jth hypothesis.

Define p_j as

$$p_j = \frac{1}{d_j}$$

and expressing it in vector form:

$$P = [p_1, p_2, \dots, p_{2^M-1}]$$

Finally, normalizing the P vector, the mass vector is generated as:

$$m_j = \frac{p_j}{\|P\|_1}, j = 1,2, \dots, 2^M - 1$$

and the mass function calculated from:

$$M = [m_1, m_2, \dots, m_{2^M-1}]$$

Probability based reasoning

Several assumptions are stipulated for this method:

1) There are N types of faults, and M features

2) All features are independent from each other

We employ initially the same formulation as in the previous section.

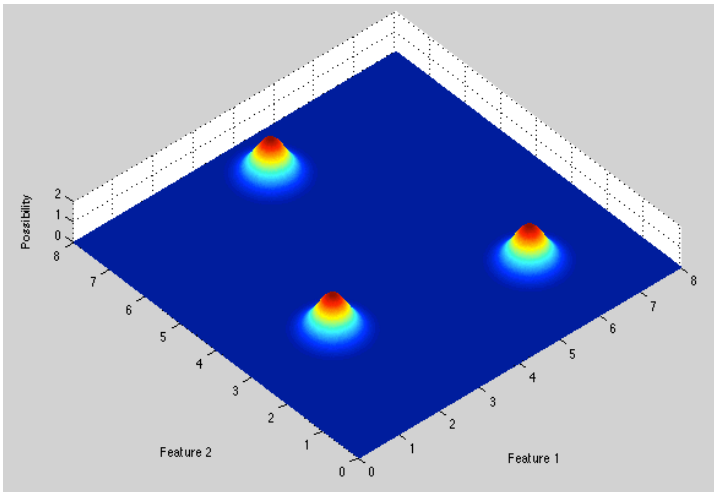


Figure 23. Distributions of fault modes

We use the existing data to fit a two-dimensional normal distribution:

$$f(x,y) = \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-\rho^2}} e^{-\frac{1}{2\sqrt{1-\rho^2}}\left[\frac{(x-\mu_x)^2}{\sigma_x^2} - \frac{2\rho(x-\mu_x)(y-\mu_y)}{\sigma_x\sigma_y} + \frac{(y-\mu_y)^2}{\sigma_y^2}\right]}$$

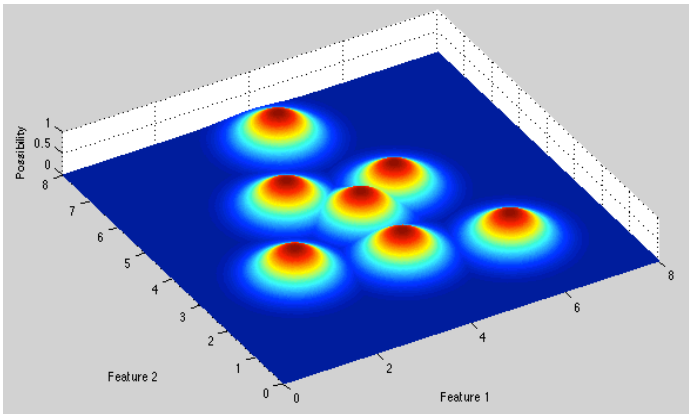


Figure 24. Distributions of hypotheses

In this case, as the two features are independent, ρ is equal to 0. So the distribution now becomes:

$$f(x,y) = \frac{1}{2\pi\sigma_x\sigma_y} e^{-\frac{1}{2}\left[\frac{(x-\mu_x)^2}{\sigma_x^2}+\frac{(y-\mu_y)^2}{\sigma_y^2}\right]} = \frac{1}{\sqrt{2\pi}\sigma_x} e^{-\frac{(x-\mu_x)^2}{2\sigma_x^2}} \cdot \frac{1}{\sqrt{2\pi}\sigma_y} e^{-\frac{(y-\mu_y)^2}{2\sigma_y^2}}$$

$$= f(x) \cdot f(y)$$

Thus, it is written as the product of two independent one-dimensional normal distributions.

For each fault mode, the histogram is generated and then a normal distribution is fitted, as shown in Figure 23. Consider next the hypotheses where multiple elements are present. For each hypothesis j we have the label vector L_j. Based on this, the distribution is generated by the following criteria:

$$\begin{bmatrix}\mu_{xj}\\ \mu_{yj}\end{bmatrix} = \frac{\begin{bmatrix}\mu_{x1} & \cdots & \mu_{xM}\\ \mu_{y1} & \cdots & \mu_{yM}\end{bmatrix} {L_j}^T}{\left\|L_j\right\|_2^2} = \frac{\begin{bmatrix}\mu_{x1} & \cdots & \mu_{xM}\\ \mu_{y1} & \cdots & \mu_{yM}\end{bmatrix} \begin{bmatrix}l_1\\ l_2\\ \vdots\\ l_M\end{bmatrix}}{\left\|L_j\right\|_2^2}$$

$$\begin{bmatrix}\sigma_{xj}\\ \sigma_{yj}\end{bmatrix} = \frac{\sqrt{\begin{bmatrix}\sigma_{x1}^2 & \cdots & \sigma_{xM}^2\\ \sigma_{y1}^2 & \cdots & \sigma_{yM}^2\end{bmatrix} {L_j}^T}}{\left\|{L_j}^2\right\|} = \frac{\sqrt{\begin{bmatrix}\sigma_{x1}^2 & \cdots & \sigma_{xM}^2\\ \sigma_{y1}^2 & \cdots & \sigma_{yM}^2\end{bmatrix} \begin{bmatrix}l_1\\ l_2\\ \vdots\\ l_M\end{bmatrix}}}{\left\|{L_j}^2\right\|}$$

Thus, all the distributions are generated as in Figure 24. For any given states, the actual state vector generated from the sensor suite is represented as: $S = [s_1, s_2, \ldots, s_n]$ As in our case, there are only two features, then the S=[x_0, y_0]. Define P in a vector form as:

$$P = [p_1, p_2, \ldots, p_{2^M-1}]$$

Each element in P is generated by the likelihood S for each distribution:

$$p_i = f_i(x_0, y_0) = \frac{1}{2\pi\sigma_x\sigma_y} e^{-\frac{1}{2}\left[\frac{(x_0-\mu_x)^2}{\sigma_x^2}+\frac{(y_0-\mu_y)^2}{\sigma_y^2}\right]}, i = 1,2, \dots 2^M - 1$$

Normalizing the P vector, the mass vector is derived by:

$$m_j = \frac{p_j}{\|P\|_1}, j = 1,2, \dots, 2^M - 1$$

$$M = [m_1, m_2, \dots, m_{2^M-1}]$$

Thus, the mass functions are generated.

For conflict resolution, we introduce the following Mean Error Bar (MEB) metric:

$$MEB = \int_{t=0}^{t_f} (Pl(t) - Bel(t))dt$$

Or, in discrete form:

$$MEB = \sum_{n=0}^{N} (Pl(n) - Bel(n))$$

As shown, the belief and plausibility functions give the lower and upper bounds of the possibility function, respectively. The value Pl(t)-Bel(t) stands for the ignorance of the possibility at time t. Usually the possibility is given by the mean of the plausibility and belief functions. If the two values are close, a precise estimate of the possibility function could be given with a small error. The MEB is, therefore, an appropriate performance metric.

Preliminary results from stored data/information suggest that both the fuzzy logic based and the probabilistic approaches are capable of addressing the conflict resolution problem. The choice of one method over the other depends on the data availability. With a limited amount of data, the fuzzy approach may reach reasonable conclusions in resolving conflicts between the pilot and the AS. With a statistically sufficient amount of data, the probabilistic method is more rigorous arriving at verifiable results.

Particle filtering based mass generation

The first step is generating the distribution for each time instant. As shown in Figure 25, after we get a new measurement the particles are resampled. The distribution at time t is: $f_x(x,t)$. The time get last sample is t_0. Then distribution $f_x(x,t), t \ll t_0$ could be used for reasoning; distribution $f_x(x,t), t > t_0$ is for prediction.

The next step is classifying the distribution into different categories of hypotheses. Here we can apply the fuzzy set structure. If we want to categorize the measured value to n levels, then there should be $2^n - 1$ hypotheses.

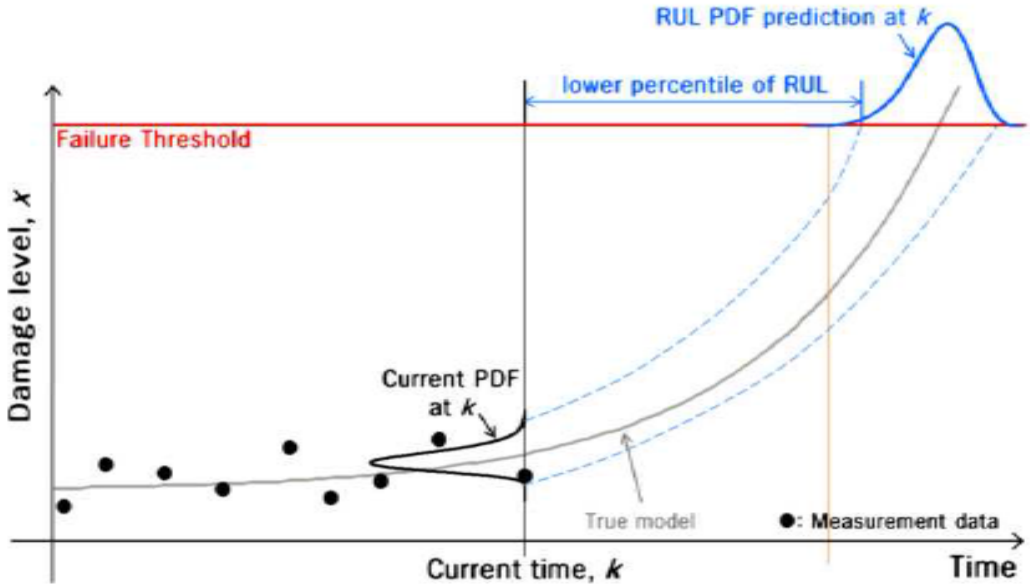


Figure 25. Typical Particle Filtering Routine

These classification functions satisfy:

$$\sum_{i=1}^{2^n-1} f_i^c(x) = 1, \text{for } \forall x$$

Then use this function set to generate the mass value for each time instant:

$$m_i(t) = \int_D f(x,t)\, f_i^c(x)dx, i = 1,2,\dots,2^n - 1$$

D is the set of x. After the mass function is generated, we can do the conflict resolution.

7.3 Game-Theoretic Framework

7.3.1 Problem Definition

We are pursuing a game-theoretic approach to conflict resolution. The proposed

framework consists of three components: the operator (pilot) and the Automated System (AS) as the players, the type of game (set as a competitive game) and the payoff or objective function. The most critical aspect of the approach is the definition of the payoff function. The solution is generally set as an optimization problem. The payoff function, for purposes of our goal, is defined as:

$$payoff_{ij} = c_j + r_{ij}$$

It consists of two terms: $c(\cdot) and\ r(\cdot)$, which represent cost and risk, respectively. The cost value depends only on the specific action taken. The risk value, on the other hand, is a function of the available information (evidence) or the lack thereof. For instance, the evidence shown to the pilot might be suggesting that he/she needs to land the aircraft immediately and he/she issues a command "land the aircraft immediately". But under the same circumstances the automated system is not exposed to the same evidence and on the basis of the information available to the AS suggests "continue flying". There is obviously a risk that the pilot is correct and the command "continue flying" may entail a significant risk. The risk function value, therefore, relies on the current status and the actions both players might take.

Player 1 (pilot) bases his/her action on the estimate of the current system status whereas player 2 (AS) suggests a strategy derived from a mix of its own evidence and the operator's estimate.

The game proceeds as follows: The pilot's action set is stated as

$$S_p = [p_1, p_2, p_3]$$

where $p_1, p_2\ and\ p_3$ represent the possible actions suggested by the operator/pilot. The conflict resolution module receives and monitors this set. Then, the Automated System will use the payoff functions to provide another mix strategy

$$S_{combined} = [q_1, q_2, q_3]$$

Thus, the probability of the Case "Pilot suggests Action i and AS recommends Action j" is expressed as:

$$p_{ij} = p_i \cdot q_j$$

And the corresponding payoff is $payoff_{ij}$. So the game is formulated as a two-step optimization problem. The first step aims to minimize the total payoff, represented by:

$$\min totalpayoff = \sum_{i=1}^{3}\sum_{j=1}^{3}(r_{ij} + c_j)\, p_{ij}$$

$$s.t \quad p_{ij} = p_i \cdot q_j \qquad for\ i, j = 1,2,3$$

$$\sum_{j=1}^{3} q_j = 1$$

Thus, by solving this optimization problem we arrive at the best $S_{combined}$, which is the advisory provided back to the pilot from the conflict resolution module. The advantage of this formulation is that it contains the information assessed by both the pilot and automated system.

Since the solution to this optimization problem might not be unique, an additional step is taken:

$$\min Stradiff = \sum_{i=1}^{3} (p_i - q_i)^2$$

$$s.t \quad [q_1, q_2, q_3] \in S_{combined}$$

Where the $S_{combined}$ set is the one derived from the first step.

The final advisory is given by solving the optimum of the following problem:

$$\min_{q_1, q_2, \cdots, q_N} E = \omega_1 \sum_{i=1}^{N} (p_i - q_i)^2 + \omega_2 [p_1, p_2, \cdots, p_N] \begin{bmatrix} PO_{11} & \cdots & PO_{1N} \\ \vdots & \ddots & \vdots \\ PO_{N1} & \cdots & PO_{NN} \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_N \end{bmatrix}$$

$$s.t \quad 0 \ll q_i \ll 1$$

$$\sum_{i=1}^{N} q_i = 1$$

Here the $p = [p_1, p_2, \cdots, p_N]$ is the estimated pilot's actions. This vector is generated by estimating the possible pilot's action. And $q = [q_1, q_2, \cdots, q_N]$ is the decision making system's advisory, which will be provided by optimize the objective function E. ω_is are the weights for these two terms. PO matrix contains the payoff coefficients.

7.3.2 Problem Analysis

To reach the minimum, we should have:

$$\frac{\partial E}{\partial q_i} = 0$$

Which equals:

$$-2\omega_1(p_i - q_i) + \omega_2 \sum_{j=1}^{N} p_j PO_{ji} = 0$$

So,

$$q_i = p_i - \frac{\omega_2 \sum_{j=1}^{N} p_j PO_{ji}}{2\omega_1}$$

If $q_i < 0$, the minimum will be set as $q_i = 0$.

It is possible that the sum of q_i does not equal to 1, so the next step is:

$$\overline{q_i} = \frac{q_i}{\sum q_i}$$

7.3.3 Sensitivity Analysis

First we can analyze the effect of Δp on Δq. Assume we have:

$$\Delta p = [\Delta p_1, \Delta p_2, \cdots, \Delta p_N]$$

Which satisfies,

$$\sum_{i=1}^{N} \Delta p_i = 0$$

So,

$$\Delta q_i = \Delta p_i - \frac{\omega_2 \sum_{j=1}^{N} \Delta p_j \, PO_{ji}}{2\omega_1}$$

$$= \Delta p_i - \frac{\omega_2 \left(\sum_{j=1}^{i-1} \Delta p_j PO_{ji} + \Delta p_i PO_{ii} + \sum_{j=i+1}^{N} \Delta p_j PO_{ji} \right)}{2\omega_1}$$

$$= \Delta p_i - \frac{\omega_2 \left[\sum_{j=1}^{i-1} \Delta p_j PO_{ji} - \left(\sum_{j=1}^{i-1} \Delta p_j + \sum_{j=i+1}^{N} \Delta p_j \right) PO_{ii} + \sum_{j=i+1}^{N} \Delta p_j PO_{ji} \right]}{2\omega_1}$$

$$= \Delta p_i - \frac{\omega_2}{2\omega_1} \left[\sum_{j=1}^{i-1} \Delta p_j \left(PO_{ji} - PO_{ii} \right) + \sum_{j=i+1}^{N} \Delta p_j \left(PO_{ji} - PO_{ii} \right) \right]$$

Now we have,

$$\frac{\Delta q_i}{\Delta p_j} = \begin{cases} -\frac{\omega_2}{2\omega_1} \left(PO_{ji} - PO_{ii} \right) & if \; i \neq j \\ 1 & i = j \end{cases}$$

$$\frac{\Delta q}{\Delta p} = \begin{bmatrix} 1 & -\frac{\omega_2}{2\omega_1} (PO_{21} - PO_{11}) & \cdots & -\frac{\omega_2}{2\omega_1} (PO_{N1} - PO_{11}) \\ -\frac{\omega_2}{2\omega_1} (PO_{12} - PO_{22}) & 1 & \cdots & -\frac{\omega_2}{2\omega_1} (PO_{N2} - PO_{22}) \\ \vdots & \vdots & \ddots & \vdots \\ -\frac{\omega_2}{2\omega_1} (PO_{1N} - PO_{NN}) & \cdots & \cdots & 1 \end{bmatrix}$$

The other aspect is a much more complicated one: error introduced by imprecision of evidence estimation. First we assume we have M key components and each component has N levels which will leads the system choose N different actions. The probability of the i^{th} component is of j^{th} level situation is assigned as ap_{ij}. So the probability of choosing action k is assigned by the following rule:

$$P_k = \sum_{\max j=k} \prod_{i=1}^{M} ap_{ij}$$

To make the calculation easier, we represent it in the following way:

$$P_k = \sum_{\max j=k} \prod_{i=1}^{M} ap_{ij} = \sum_{1 \le j \le k} \prod_{i=1}^{M} ap_{ij} - \sum_{1 \le j \le k-1} \prod_{i=1}^{M} ap_{ij}$$
$$= \prod_{i=1}^{M} \sum_{j=1}^{k} ap_{ij} - \prod_{i=1}^{M} \sum_{j=1}^{k-1} ap_{ij}$$

This step will reduce the calculation complexity greatly, because instead of the N^M time loop, this algorithm just needs calculate several addition and multiplication.

Our risk table is a big table with the following structure:

Table 4. Structure for risk table

		Action 1	Action 2	…
Component 1	Level 1	r_{111}	r_{112}	
	Level 2	r_{121}		
	⋮			
Component 2	Level 1	r_{211}	r_{212}	
	Level 2			
	⋮			

Here we define the notation r_{ijk}, which means the risk of automated system suggests component i is at the condition j and the pilot believes they should take action k. So we know,

$$r_{ijk} = 0$$

for all $j \le k$.

Then the risk of automated system suggest action m and pilot claims taking action n should be:

$$R_{mn} = \frac{\sum_{\max j=m}(\prod_{i=1}^{M} ap_{ij} \cdot \sum_{i=1}^{M} r_{ijn})}{\prod_{i=1}^{M}\sum_{j=1}^{m} ap_{ij} - \prod_{i=1}^{M}\sum_{j=1}^{m-1} ap_{ij}}$$

Of them the r_{ijk} are constants that will be evaluated based on old experience. ap_{ij} is evidences estimated by the automated system, which might introduce error.

And for the cost function, there will be another cost table. So we have:

$$PO_{mn} = R_{mn} + C_{mn}$$

Here the C matrix is a constant table. So there will not be any disturbance introduced in this term. As we described before:

$$q_i = p_i - \frac{\omega_2 \sum_{j=1}^{N} p_j \, PO_{ji}}{2\omega_1}$$

So we have:

$$\Delta q_n = -\frac{\omega_2 \sum_{m=1}^{N} p_m \, \Delta PO_{mn}}{2\omega_1} = -\frac{\omega_2 \sum_{m=1}^{N} p_m \, \Delta R_{mn}}{2\omega_1}$$

$$= -\frac{\omega_2}{2\omega_1}\sum_{m=1}^{N} p_m \frac{\sum_{\max j=m}(\sum_{i=1}^{M} r_{ijn} \cdot \prod_{i=1}^{M} \Delta ap_{ij})}{\prod_{i=1}^{M}\sum_{j=1}^{m} ap_{ij} - \prod_{i=1}^{M}\sum_{j=1}^{m-1} ap_{ij}}$$

7.4 Information Fusion Algorithm

Now we discuss when the two assessments from the pilot and the automated system agreed with each other. Assume we have the pilot's suggestion as:

$$S_p = [p_1, p_2, \cdots, p_N]$$

Similarly we assume the Automated system suggest the action should be:

$$S_q = [q_1, q_2, \cdots, q_N]$$

Here we know all the p's and q's are positive and less than 1. When they agree with each other, the combined result is defined as:

$$S_{combined} = \left[\frac{p_1 q_1}{\sum_{i=1}^{N} p_i q_i}, \quad \frac{p_2 q_2}{\sum_{i=1}^{N} p_i q_i} \quad , \cdots, \frac{p_N q_N}{\sum_{i=1}^{N} p_i q_i}\right]$$

Assume the Action i is the suggestion from the pilot and the automated system, then we know we have:

$$p_i > \forall p_j \, , for \; j \neq i$$

$$q_i > \forall q_j \, , for \; j \neq i$$

Thus,

$$p_i q_i > \forall p_j q_j \, , for \; j \neq i$$

So the Action i is still the suggestion from the combined result with a confidence level of:

$$conf_i = \frac{p_\mathrm{i} q_\mathrm{i}}{\sum_{j=1}^{\mathrm{N}} p_j q_j}$$

We also can prove $conf_i$ is larger than p_i or q_i:

$$conf_i - p_i$$

$$\left\{\sum_{j=1}^{\mathrm{N}} p_j = 1\right\} = \frac{p_\mathrm{i} q_\mathrm{i}}{\sum_{j=1}^{\mathrm{N}} p_j q_j} - \frac{p_i}{\sum_{j=1}^{\mathrm{N}} p_j}$$

$$= \frac{p_i q_i \sum_{j=1}^{N} p_j - p_i \sum_{j=1}^{N} p_j q_j}{\sum_{j=1}^{N} p_j q_j \cdot \sum_{j=1}^{N} p_j}$$

$$= \frac{p_i q_i \left(p_i + \sum_{j \neq i} p_j\right) - p_i \left(p_i q_i + \sum_{j \neq i} p_j q_j\right)}{\sum_{j=1}^{N} p_j q_j \cdot \sum_{j=1}^{N} p_j}$$

$$= \frac{p_i \sum_{j \neq i} p_j (q_i - q_j)}{\sum_{j=1}^{N} p_j q_j \cdot \sum_{j=1}^{N} p_j}$$

As we have:

$$q_i > \forall q_j \, , for \; j \neq i$$

So,

$$conf_i - p_i > 0$$

Similarly we can prove:

$$conf_i - q_i > 0$$

So we know the combined result has a higher confidence level than the assessment of either player.

CHAPTER 8

STRUCTURE WITH FEEDBACK AND TIME SEQUENCES

8.1 Introducing the Feedback Structure

Decision must be maximally sensitive to pilot and AS's inputs, and minimally sensitive to noise and disturbances. Assume noise or disturbance profile additive or multiplicative, then how can we reduce its influence to decision making? So it is reasonable to introduce feedback to our system.

Consider decision-making process as a dynamic system with uncertainty and delays. The original structure is shown in Figure 26.

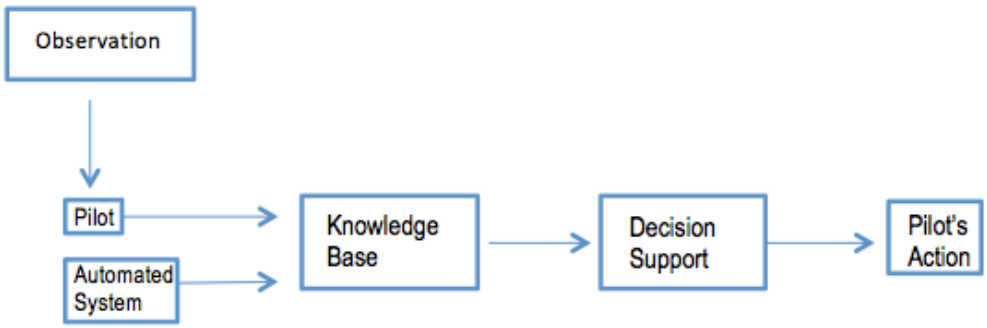


Figure 26. Original Schematic

Obviously it is an open loop system, so we can add a **Success or Failure** module as the feedback, to close the loop. The Success or Failure module is used to estimate if

the pilot's actions really has positive effect on the condition of the aircraft and flying. For example, the suggested action is "Landing preparation-Engine Maint." Then the pilot takes the action and triggered the following monitoring routines:

1) The pilot can land the aircraft safely within 500km following provided path
2) Engine parameters lower than hazard zone 1
3) Other parameters within normal zone
4) Outside parameters-OK

If one of the above criterions is checked as "False", then the system decides the action is failed. Otherwise, the system believes the action is success.

The revised version of structure (containing Success or Failure Module) is shown in Figure 27.

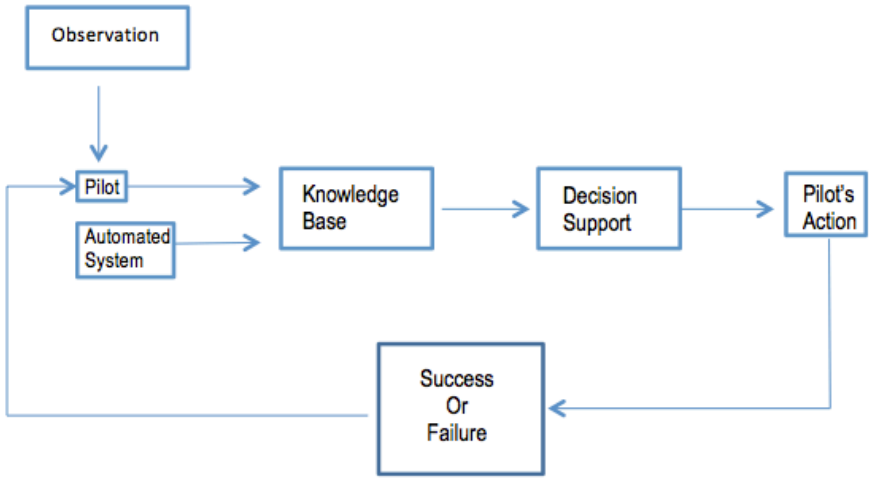


Figure 27. System Schematic with Feedback

8.2 Output of Success or Failure Module

So in the updated structure, the pilot will be provided the output of Success or Failure Module. If the judgment is “Success”, then the pilot will be given information like “Action Provided Success”. Otherwise, the system will provide the details of why the suggested actions are failed. For instance, the advisory by the decision-making system is “Landing Preparation-Engine Maint.” After the pilot execute the advisory, the Success of Failure module will say the following:

Action Provided -- Failed

Engine Condition -- higher than hazard zone 1

IGB -- higher than normal zone

Then based on the feedback of Success or Failure module the pilot can use the provided tuning matrix T to tune the former action vector:

$$\widetilde{P}_{cb} = T \cdot P_{cb}$$

Here the P_{cb} is the action vector provided by the decision-making system, the combined result. $\widetilde{P}_{cb}$ is the action vector after adjusted. So when the advisory is successful, T will be assigned as identity matrix I. For cases like the failure case we described above, then we have:

$$T = \begin{bmatrix} 0.9 & 0 & 0 \\ 0.1 & 1 & 0.1 \\ 0 & 0 & 0.9 \end{bmatrix}$$

Which means:

$$\tilde{p}_1 = 0.9p_1$$
$$\tilde{p}_2 = 0.1p_1 + p_2 + 0.1p_3$$
$$\tilde{p}_3 = 0.9p_3$$

So in general, the matrix T has the following properties

- When the provided advisory is successful, $T = I$
- $0 \leq T_{ij} \leq 1$
- Because

$$\sum \tilde{p}_i = 1$$

So we have

$$\sum_{i=1}^{N} \tilde{p}_i = \sum_{i=1}^{N} \sum_{j=1}^{N} p_j \cdot T_{ij} = \sum_{j=1}^{N} (\sum_{i=1}^{N} T_{ij}) \cdot p_j$$

As we know

$$\sum p_j = 1$$

So we can assign

$$\sum_{i=1}^{N} T_{ij} = 1$$

Of course we can apply some other ways to assign the value of the T matrix, but obviously it may lead the algorithm more complicated and might encounter following problems:

1. $\tilde{p}_i$ might be negative
2. $\sum \tilde{p}_i \neq 1$, thus additional normalizing step is needed.

So the rules above can guarantee the adjusted suggestion vector is reasonable and avoid additional calculation.

8. 3 New Expression of the structure

Now we can express the system as an adaptive system. The new structure, with the pilot in the center, is shown in Figure 28.

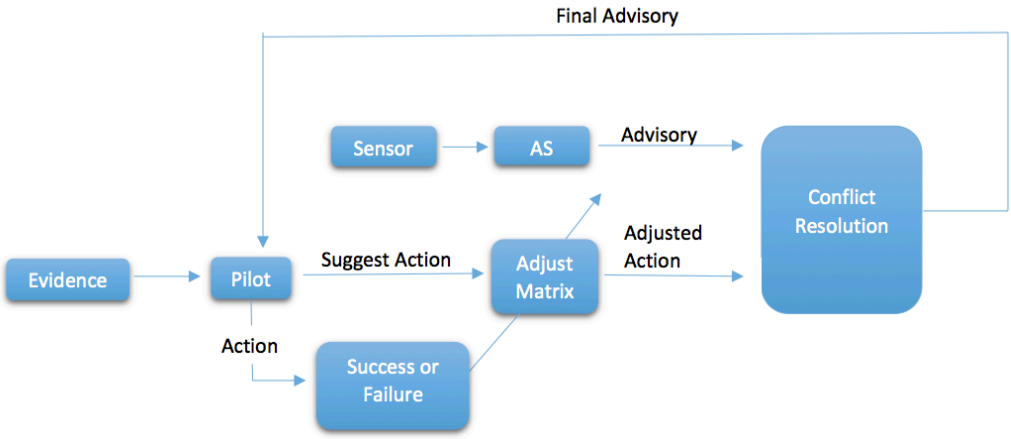


Figure 28. Pilot Centered Structure with Adaptive Control

Here the success or failure module does not return the result to the pilot, but directly adjust the pilot's estimation. The advantage of this structure is the pilot does not have to process too much information and can avoid problems of information overflow.

An alternative way to deal with this problem is sending the output of Success or

Failure Module directly to the pilot. Thus this output will act as another resource of pilot's evidence. We notice that right now the pilot is no longer equivalent to the automated system because the pilot will access more information source and has more output instead of only the pilot's suggested action. So it is reasonable to generate another pilot-centered structure, as shown in Figure 29. In the figure around the pilot module, there are two information sources: Evidences and Advisory source, which are represented by solid line and dashed line respectively. The pilot uses the evidences and S/F result to generate the suggested action, and collect the decision making system's final advisory to come up with the action to execute.

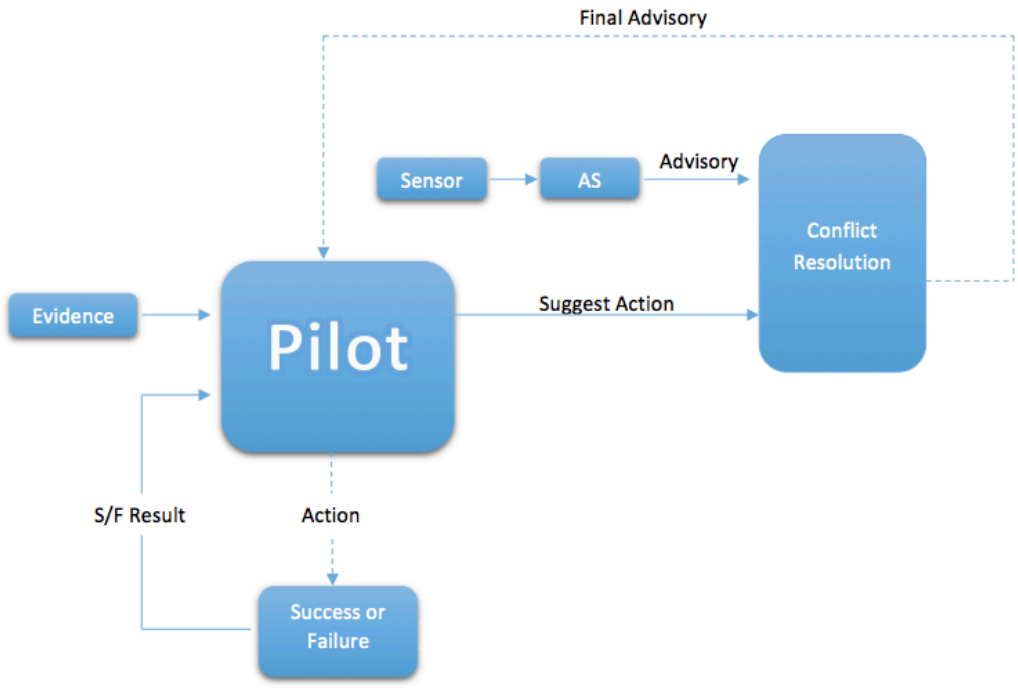


Figure 29. Pilot Centered Structure with S/F Module as a Feedback Module

With delays considered, the pilot centered architecture is represented in Figure 30. In this schematic the delays are represented by the blocks e^{-ps}, p is the coefficient

related to delay.

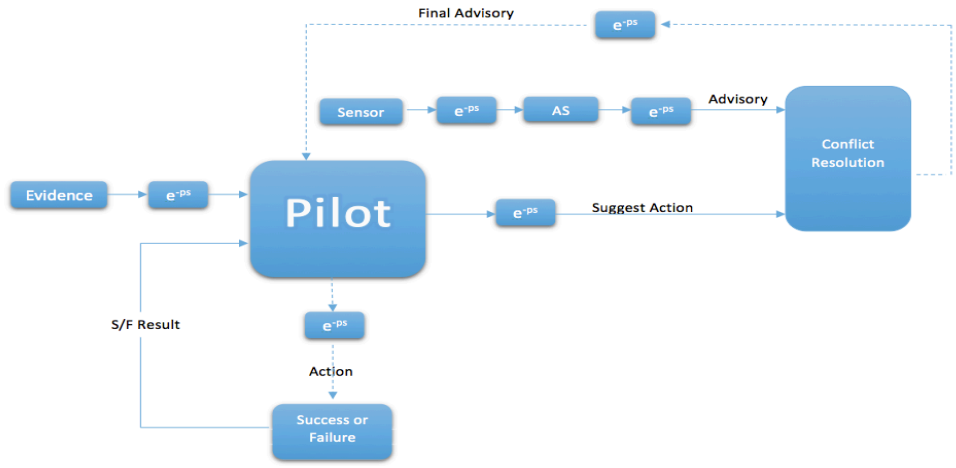


Figure 30. Pilot Centered Structure with Continuous Time Delay Blocks

Or discrete form,

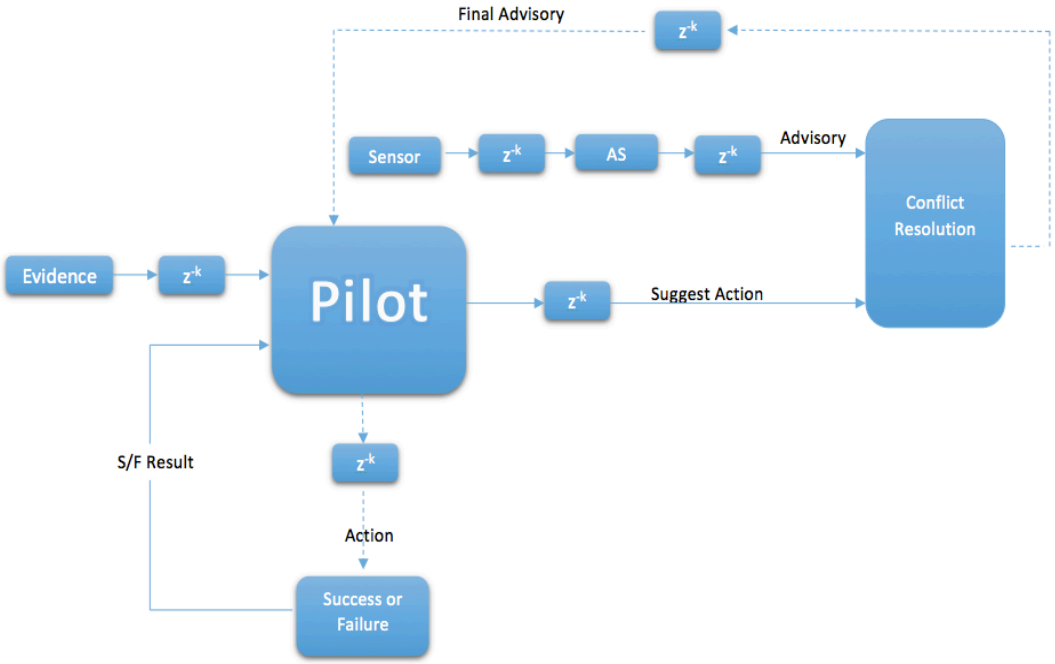


Figure 31. Pilot Centered Structure with Discrete Time Delay Blocks

8.4 Time Sequences Analysis

The pilot-automated system interface introduced in this thesis is viewed as a dynamic system of systems that may be evolving as time progresses and both the human (pilot) and the automated system are subjected to new events, evidences or data/information while the vehicle is obeying the pilot's commands. With the assumption that the decision making process is relatively fast compared with the vehicle's dynamic performance it may be possible to track the conformance of the vehicle's behavior to the commanded actions while, at the same time, monitor and assess any changes in the data or observations both the pilot and the automated system are basing their corresponding decisions/advisories.

In that case it is advisable to view the overall system configuration as a dynamic system and exploit any available new information to update the conflict resolution and decision support processes. We introduce in this chapter system architecture to achieve this task.

Current and past research in this area has considered the human-system interface primarily as a once-through strategy for decision making problem, i.e. the decision making process is executed once and the pilot follows the advisories generated from a combination of on-board data/information and his/her own observations/experience. As shown in Figure 32, in phase I, the pilot collects his/her evidences by making observation or judging from his experience, and come up with his suggested action. At the same time, the automated system will collect the sensor data, do the reasoning and come up with its advisory. During

this period the knowledge base will hold and wait for the pilot's suggestion and automated system's advisory coming in. In phase II, the automated system and the pilot will hold for the knowledge base to fuse the two advisories and resolve the potential conflicts. After the conflict is resolved, the knowledge base will come up with a final advisory and send this advisory to the pilot. Then phase III is when the pilot reads this advisory and takes the corresponding action. Then this routine stopped.

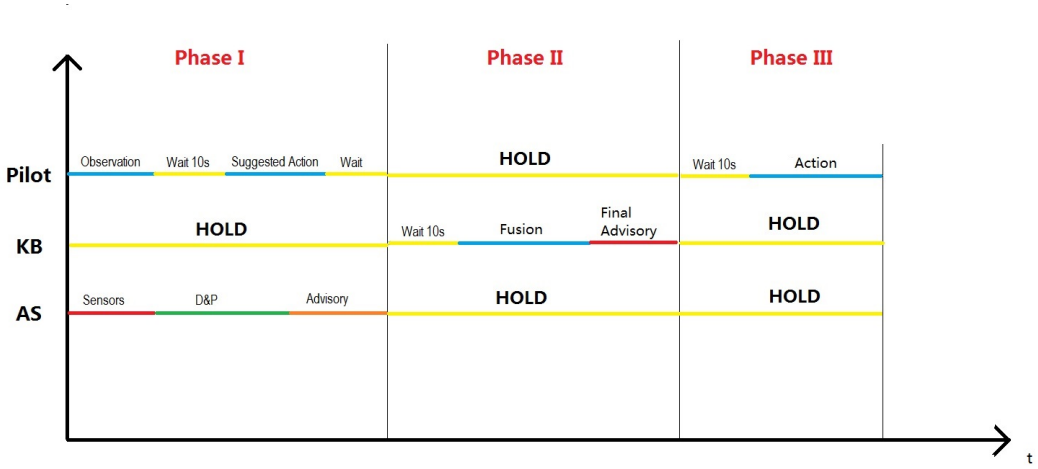


Figure 32. Once Through Time Chart

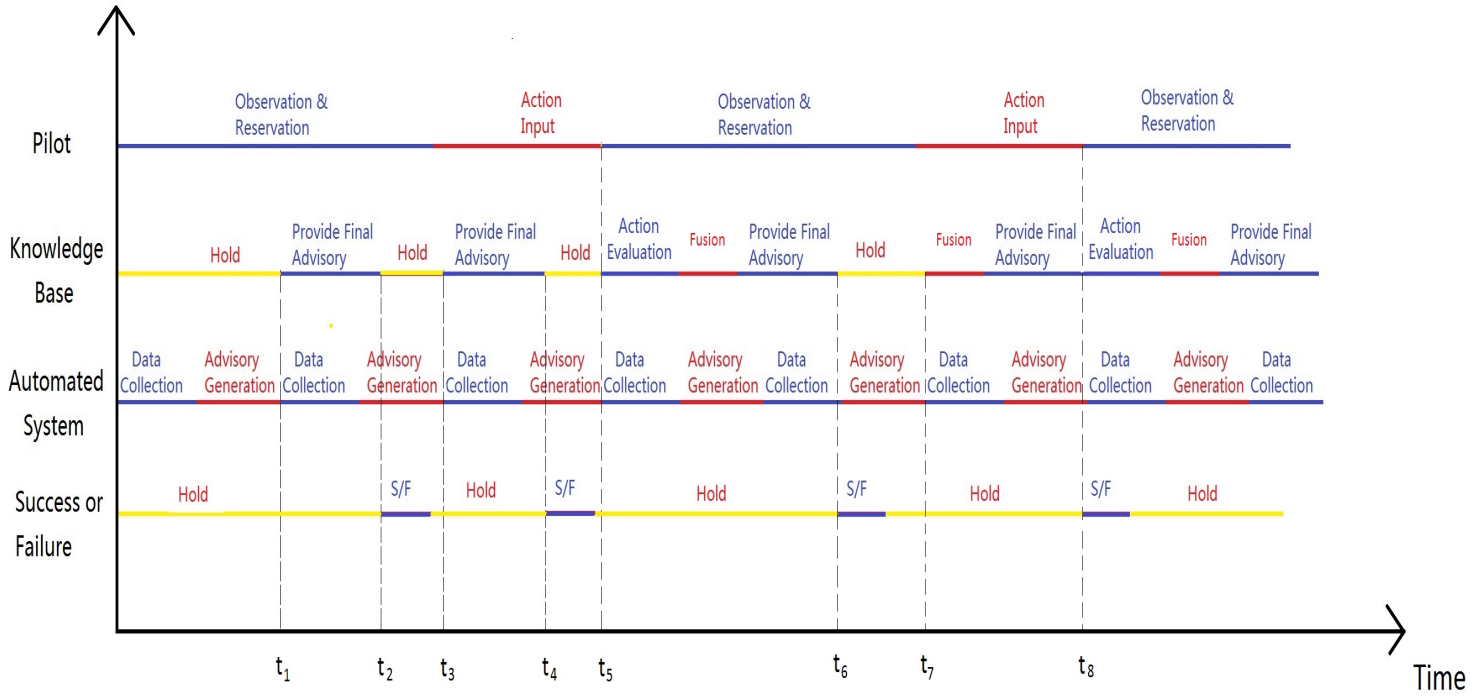


Figure 33. Dynamic Structure Time Chart

The advent of new and expedient hardware and software allows for reconsideration of this problem and the introduction of tools/methods that view the system in its dynamic formulation enabling the continuous updating of the advisories on the basis of new data/information. So after the original stop after phase III, the procedure will act repeatedly, as shown in the Figure 33.

In this structure, the main property is the pilot's reasoning and the automated system's condition estimation will act repeatedly, as shown in the first and third row. Now the automated system and the pilot will not hold for the knowledge base to resolve conflict. Pilot and automated system keep updating the system's condition, so in this structure the system does not have phases like shown in the Figure 32. Instead, we will show the structures based on the periods shown in the Figure 33.

Time 0: The beginning of the procedure

Time t_1: At this time the pilot has not finish the reasoning and action input, but the automated system has finished the reasoning and put the action to the automated system. As there's only one player put his action into the knowledge base, the knowledge base does not have to resolve potential conflict. So the knowledge base just has to hold and tell the system the final advisory is not generated. The success or failure decision is evaluated based on the final advisory, so the success or failure module will also hold.

Time t_2: At this moment, the pilot finished the action input. The knowledge base is activated. The final advisory is generated and the Success or Failure module is

activated. After the S/F procedure, the advisory is success or not will be determined.

Time t_3: The most updated automated system's advisory is provided to the knowledge base, so another conflict resolution is triggered. Pay attention that now the knowledge base has to resolve the potential conflict for each advisory generation because the pilot's action is no longer zero. But the knowledge base does not have to estimate the pilot's action again because it is not updated yet.

Time t_4: At this moment the pilot's suggestion is changed, so the knowledge base has to re-estimate the pilot's action. The other tasks are similar to the former cycles.

CHAPTER 9

IMPLEMENTATION TOOLS: GRAPHICAL USER INTERFACE (GUI)

9.1 General Structure

A Graphical User Interface (GUI) using MATLAB® has been developed to demonstrate the structural aspects of the interface, shown in the Figure 34. The structure consists of the following components: Pilot's Input Panel, Estimated Action Panel, Control Panel, Advisory Panel, Success or Failure Panel and Dynamic Condition Monitor. We introduce each panel in the sequel.

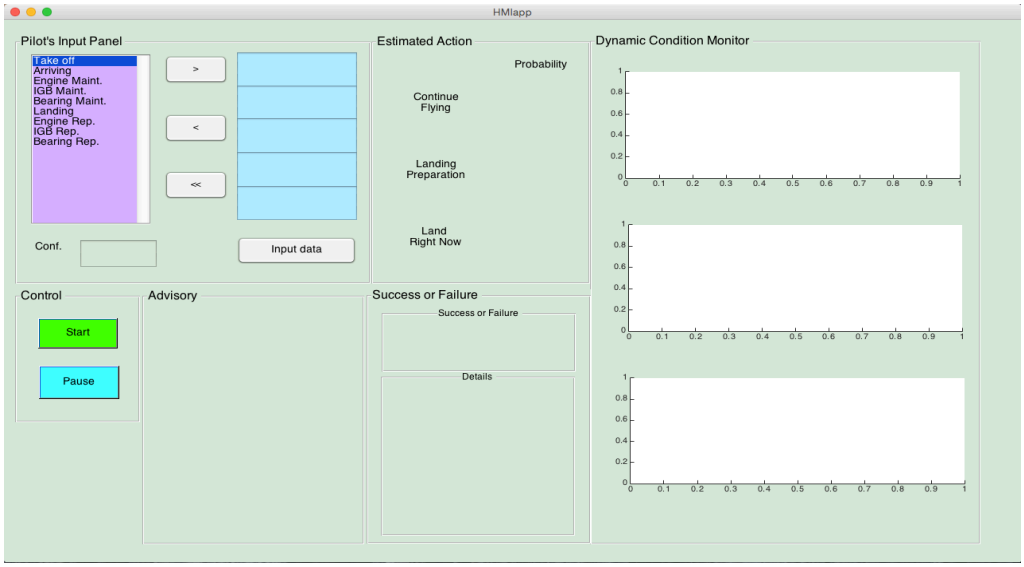


Figure 34. Graphical User Interface developed in MATLAB

The Pilot's Input Panel is the area where the pilot can change the action

sequence to be executed and the corresponding confidence level. The pilot can choose the preferred action from the action list (purple area) and add it to the current sequence, shown by the blue area. Three buttons are located between the action list and the sequence: add, delete and clear the sequence, respectively. The pilot can type in the estimated confidence in the confidence box and press the “Input Data” button to insert all the information into the knowledge Base.

The “smart” knowledge base will match now the sequence and the confidence level with stored cases. Next, the knowledge base provides an estimate of the pilot’s action, shown in the “Estimated Action Panel”. The probability of the pilot taking one of the actions ”Continue Flying”, ”Landing Preparation” and “Land Right Now” are 87.5%, 6.5% and 6%, respectively.

On the left down area is the “Control Panel”. This area contains two buttons: Start and Pause. When the Start button is pressed, the simulation routine is initiated and the start button is disabled. If the pilot wants to pause the routine, the pause button is pressed and the simulation stops for several seconds.

On the right of the figure is the “Dynamic Condition Monitor Area”. The three plots demonstrate the estimated pilot’s action, automated system’s advisory and the final advisory, respectively. Blue, red and yellow lines indicate the confidence level of “continue flying”, “landing preparation” and “land right now”. As the final advisory is the result generated by the optimization routine, this advisory is often a pure strategy instead of the common mix strategy.

To assist the pilot to easily access an executable advisory, the Advisory Section will provide a literal message. Basically it selects the action with highest possibility, expected to have the lowest payoff. Here, instead of determining the minimum value, we apply the idea of magnetic hysteresis. Details are shown in the following sections.

The last section is the “Success or Failure Section”. It consists of two sub sections: “Success or Failure” and “Detail”. The Success or Failure area will return a corresponding message when the advisory results in a successful operation or not. If it succeeds, then no further detail is required. Otherwise, the system will return a note as to which area is the abnormal one and needs special attention. This additional information is shown in the “Detail Area”.

9.2 Final Advisory Generation

Based on the optimization algorithms described previously, the basic idea of generating the final advisory is to choose the term with the minimum coefficient. The advisory is illustrated in Figure 35.

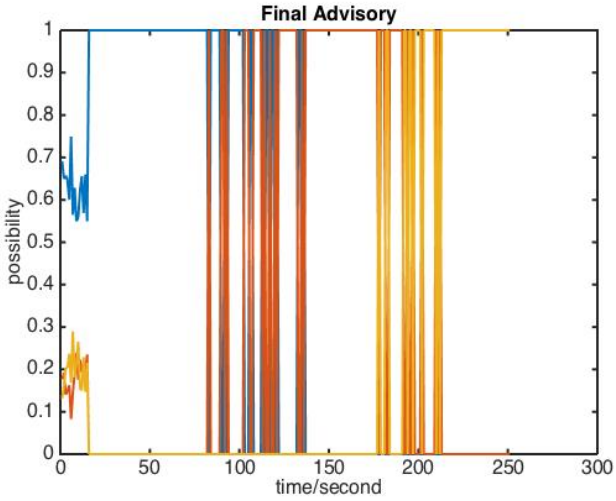


Figure 35. Original Final Advisory

Obviously this advisory is difficult to apply since frequent changes occur between two time periods due to two coefficients being in close proximity to each other resulting in the pilot's inability to apply the given advisory.

To avoid this circumstance we take advantage of the magnetic hysteresis effect and redesign the criteria as:

$$Advisory[k] = \begin{cases} Action_i[k], & if\ coeff_j[k] - coeff_i[k] > 2 \\ Advisory[k-1], & Otherwise \end{cases}$$

where

$$i \in \{i | coeff_i[k] \leq \forall coeff_m[k], 1 \leq m \leq N\}$$

$$j \in \left\{j \middle| coeff_j[k] \leq \forall coeff_q[k], 1 \leq q \leq N\ and\ q \neq i\right\}$$

We apply this algorithm on the same dataset and generate a new plot as in Figure 36.

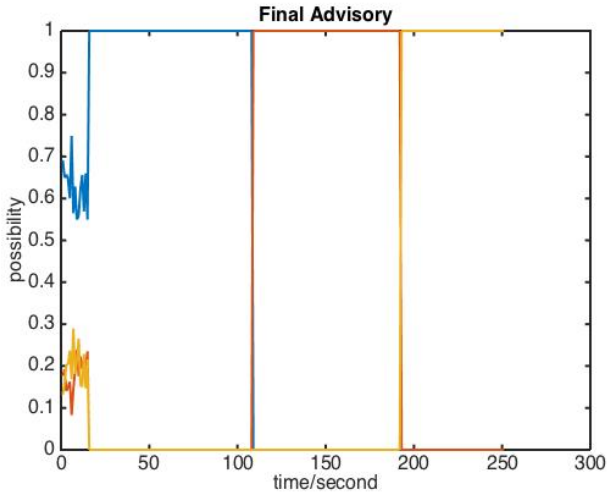


Figure 36. Filtered Final Advisory

Compared to the former algorithm, this new one results in a much more stable performance generating a feasible advisory for the pilot.

9.3 Success or Failure Module and Time Delays

It is evident that time delay is a key factor affecting the system's performance. It is important, therefore, to analyze the delays in each component of the interface. The structure, including delay blocks, is shown in a former section. There are several key delays that must be considered:

Evidence Collection and Decision Making Delay (D_1)

It is known that the time for the pilot to collect evidences and arrive at a decision varies from person to person. Also, even for the same pilot, this time can be severely decreased after the pilot is getting more familiar with the system condition and the environmental effects impacting the decision making process

and the time required for this step may vary significantly.

Another important factor is the time delay due to the time required to input the sequence into the system. This time also varies from person to person. 10 volunteers were picked to execute this task with each person tested 20 times resulting in the distribution in Figure 37.

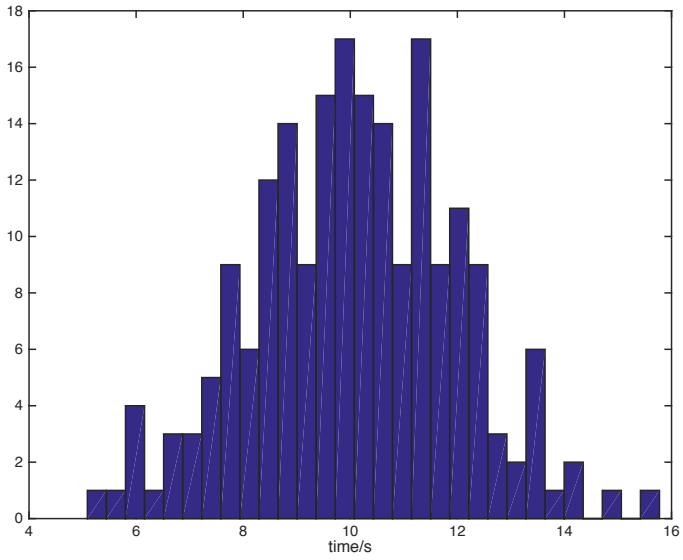


Figure 37. Distribution for D_1

The mean value of this time is 10.102 seconds. The worst case is 15.851 seconds. Being on the scale of 10 seconds, it is reasonable to assume

$$D_1 = z^{-k_1}, k_1 \sim O(10)$$

Pilot Action Evaluation Delay (D_2)

This is a delay caused by alternating the sequence in the estimated pilot's action.

The time is shown in Figure 38.The mean value of this delay is 0.0436 seconds. The worst case is 0.0723 seconds. They are on the scale of 0.01 seconds.

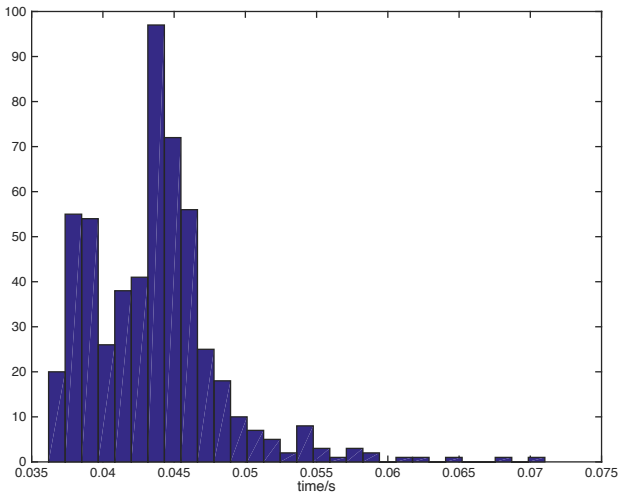


Figure 38. Distribution for D_2

So, we have:

$$D_2 = z^{-k_2}, k_2 \sim O(10^{-2})$$

Automated System reasoning Delay (D_3, D_4)

This delay is illustrated in Figure 39. The mean value of this delay is 2.09×10^{-5} seconds. The worst case is 8.48×10^{-5} seconds being on the scale of 10^{-5} seconds.

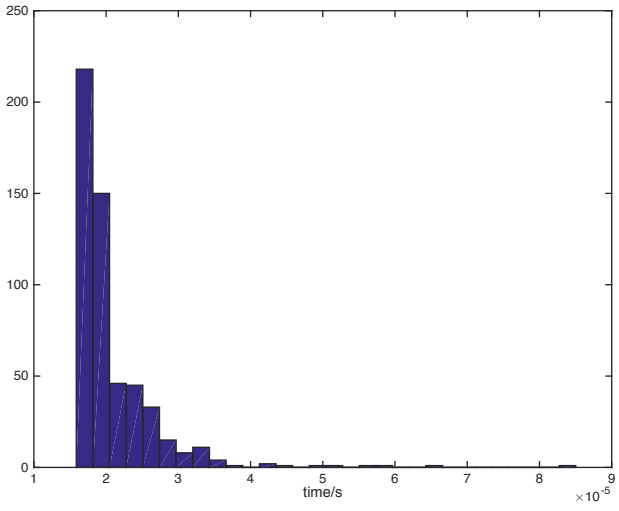


Figure 39. Distribution for D_3

Thus:

$$D_3 = z^{-k_3}, k_3 \sim O(10^{-5})$$

Conflict Resolution Delay (D_5)

The conflict resolution delay corresponds to the time required to resolve potential conflicts. The tested value is demonstrated in Figure 40.

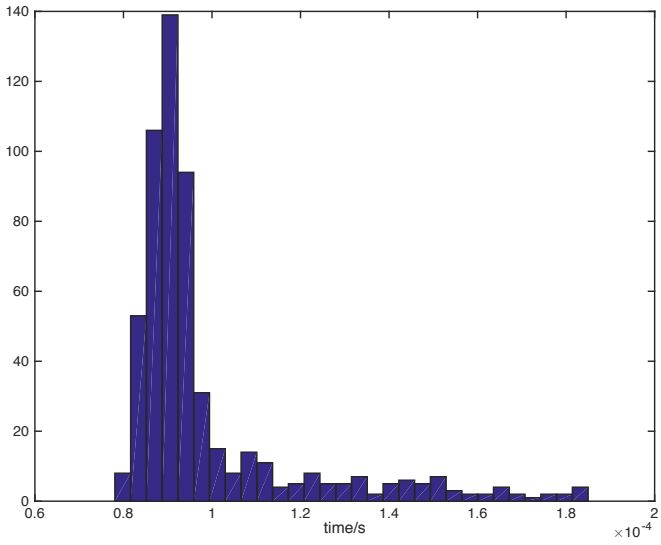


Figure 40. Distribution for D_5

The mean value of this delay is 9.89×10^{-5} seconds and thus:

$$D_5 = z^{-k_5}, k_5 \sim O(10^{-5})$$

Pilot's Execution delay (D_6)

This value also varies from person to person. Here we assume:

$$D_6 = z^{-k_6}, k_6 \sim O(1)$$

In summary, the major time delay is attributed to the time required to implement the pilot's reasoning and the time required to input it into the decision into the system, with all other delays considered as lower order.

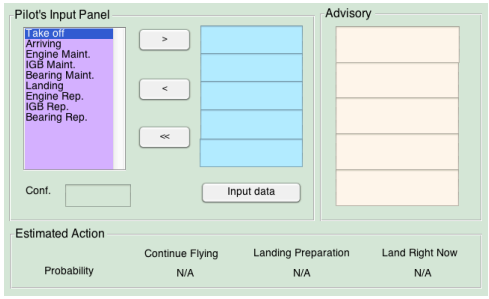
CHAPTER 10

RESULTS

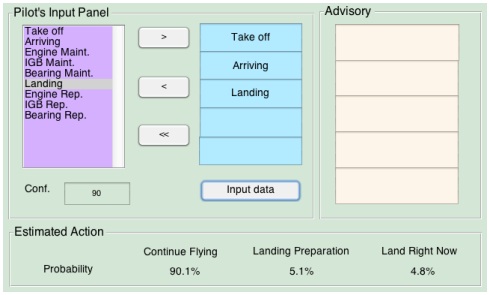
10.1 Pilot's Action Estimation

The simulated Graphical User Interface (GUI) is generated in MATLAB. The sections related to the pilot's actions are depicted in the pilot's input panel section as estimated actions. Figure 41 shows the sequence of estimated actions. In Figure 41(a), no pilot's input is received, and the estimated pilot's actions are displayed as "N/A". Later the input sequence is set as [Take off, Arriving, Landing], as shown in Figure 41(b). This implies that the pilot believes there is no fault, and the aircraft can continue flying. The estimated action indicates the probability of "continue flying" is more than 90%.

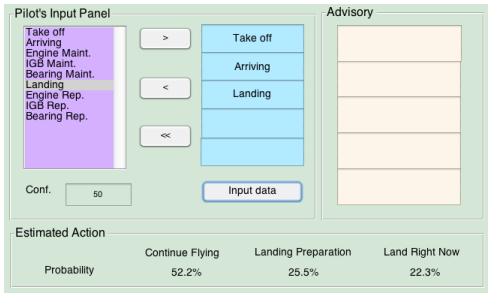
When the pilot senses something abnormal, his confidence is decreased to 50% and the estimated probability of "continue flying" decreases also to 52.2%, as shown in Figure 41(c). At that instant, the pilot suggests that the aircraft should undergo "Engine Maintenance" after it arrives to its destination. So, the estimated probability of "landing preparation" rises to 53.3%, as in Figure 41(d).



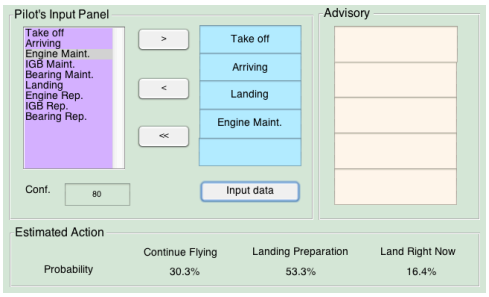
(a)



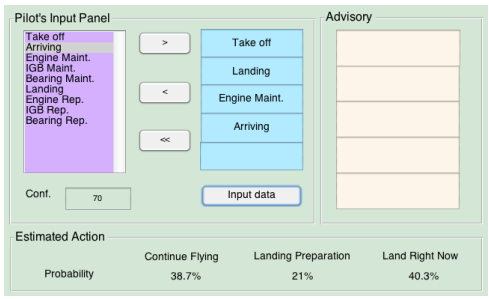
(b)



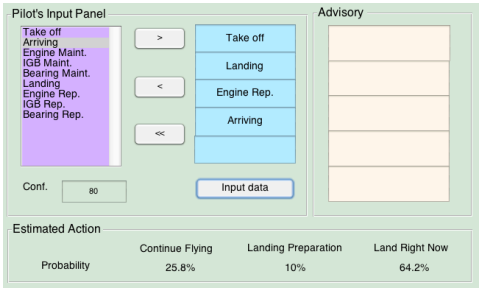
(c)



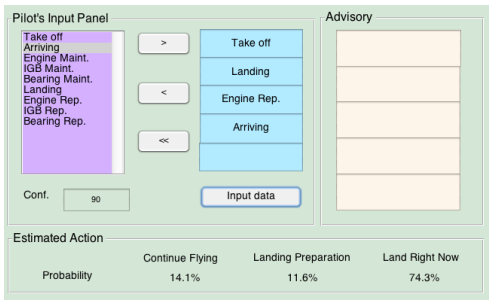
(d)



(e)



(f)



(g)

Figure 41. Multiple Stages for Pilot's Intended Action Estimation

If the evidence suggests that the situation is more severe, the pilot believes that it is necessary to conduct maintenance before the aircraft arrives at its destination. Now, safety has a higher priority than arriving at the destination and the estimated probability of "landing right now" rises to 40%, as shown in Figure 41(e). Since the condition is very severe, the pilot changes his action from "Engine Maintenance" to "Engine Repair". The estimated probability of "landing right now" rises to 64.2% and 74.3%, with respect to the confidence levels of 80% and 90%, as shown in Figure 41(f) and Figure 41(g).

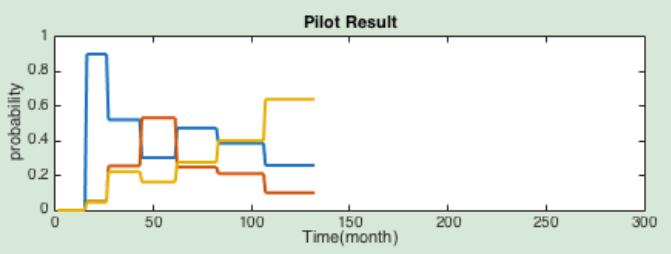


Figure 42. Initial Estimated Pilot's Intended Action

The complete procedure is depicted in Figure 42. The probabilities of "continue flying", "landing preparation" and "land right now" are shown in blue, red and yellow, respectively.

It is considered also that the pilot might accidentally input the wrong actions. The structure is very sensitive to this situation and instead of the normal

$$Action_{Pilot} = Pilot[n]$$

The following will be applied

$$Action_{Pilot} = \frac{1}{8}Pilot[n-3] + \frac{1}{8}Pilot[n-2] + \frac{1}{4}Pilot[n-1] + \frac{1}{2}Pilot[n]$$

This filter will assist to smooth the pilot's action sequence. Also, considering the uncertainty of the variable "confidence", we introduce an uncertainty value in the confidence estimation and the final estimated pilot's action is shown in Figure 43.

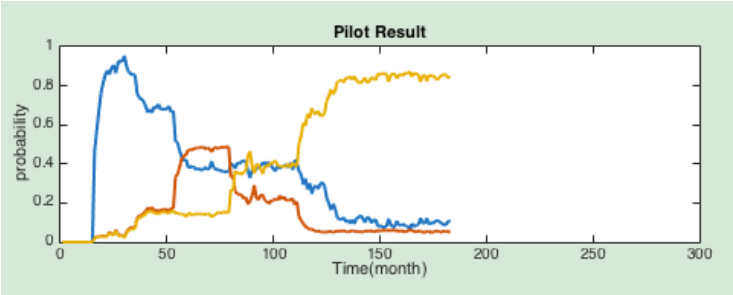


Figure 43. Filtered Pilot's Intended Action

10.2 Dempster-Shafer Theory Based Methodology Result

10.2.1 The Application Domains: Helicopter Drive System

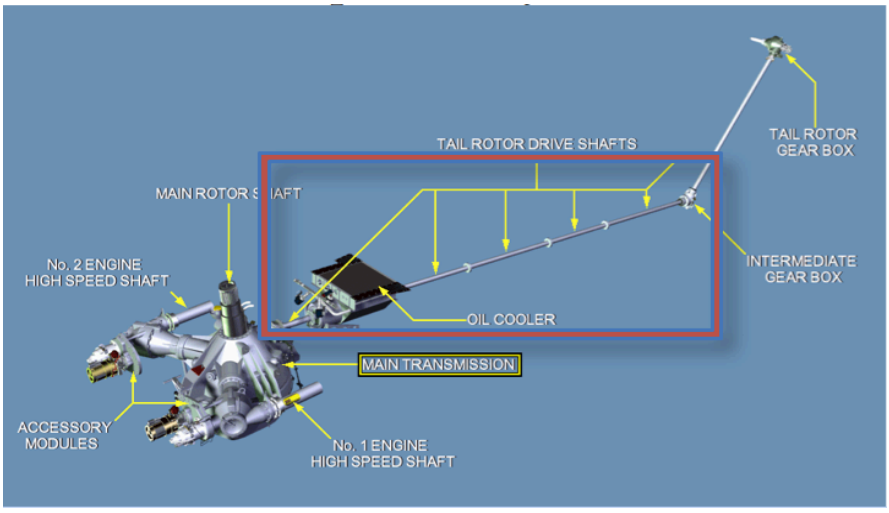


Figure 44. Drivetrain of the UH-60 Helicopter

The application domain (in simulation) for the conflict resolution configuration is the Oil-cooler & Intermediate Gearbox (OC-IGB) subsystems of the UH-60 helicopter drive system. The complete drivetrain is shown in Figure 44. The OC-IGB subsystem is highlighted by the red rectangular area. The components include the oil-cooler, the intermediate gearbox, and the tail shaft connecting these components. We define appropriate fault modes and suggest data/observations/displays available to the operator (pilot). On the other hand, we configure the automated system to accomplish sensor data collection and analysis including the diagnostic, prognostic and control modules introduced previously.

10.2.2 Simulation Results

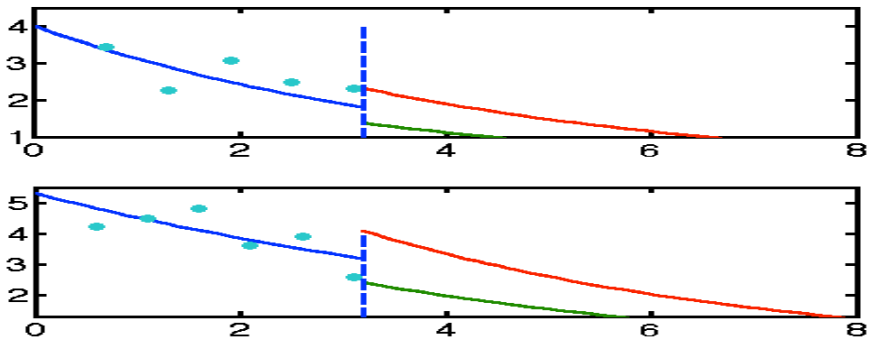


Figure 45. RUL predicted by the pilot and Automated System

The data used in this case study is generated by a MATLAB routine. The features discussed above and the status evaluations are extracted from the data set. The pilot's judgment is based on his perception while the Automated System collects the pre-processed data and provides the advisories. Then, the decision support system

reads the estimations and gives the combined reasoning result. The simulation procedure is also carried out in MATLAB.

Oil Cooler Bearing Crack Level Prognosis

The pilot and the automated system can both do the prognosis based on the information they collected. For instance, in our case, the pilot and the automated system can collect information from time 0 to time 3.2. And based on these information to predict the 3.2 to 8 system situation, as shown in Figure 45. The upper figure is generated by the pilot and the lower one belongs to the automated system. So the pilot predicts that the remaining useful life should be [4.5 6.5] with a confidence level of 90%. On the other hand, the automated system believes that remaining useful life should be [5.7 7.6] with a confidence level of 90%. Here comes the conflict between the two reasoning route. So we apply the conflict resolution here to get a combined result, as shown in Figure 46. Thus the system status can be got from the combined result. For instance, oil bearing crack level for time 5 should be light or medium with a confidence level of 60%. Also we can calculate the MEB and get Table 5. The table illustrate that the combined result has much smaller MEB than the pilot or AS, which means the combined result reduces the risk, or ignorance, significantly.

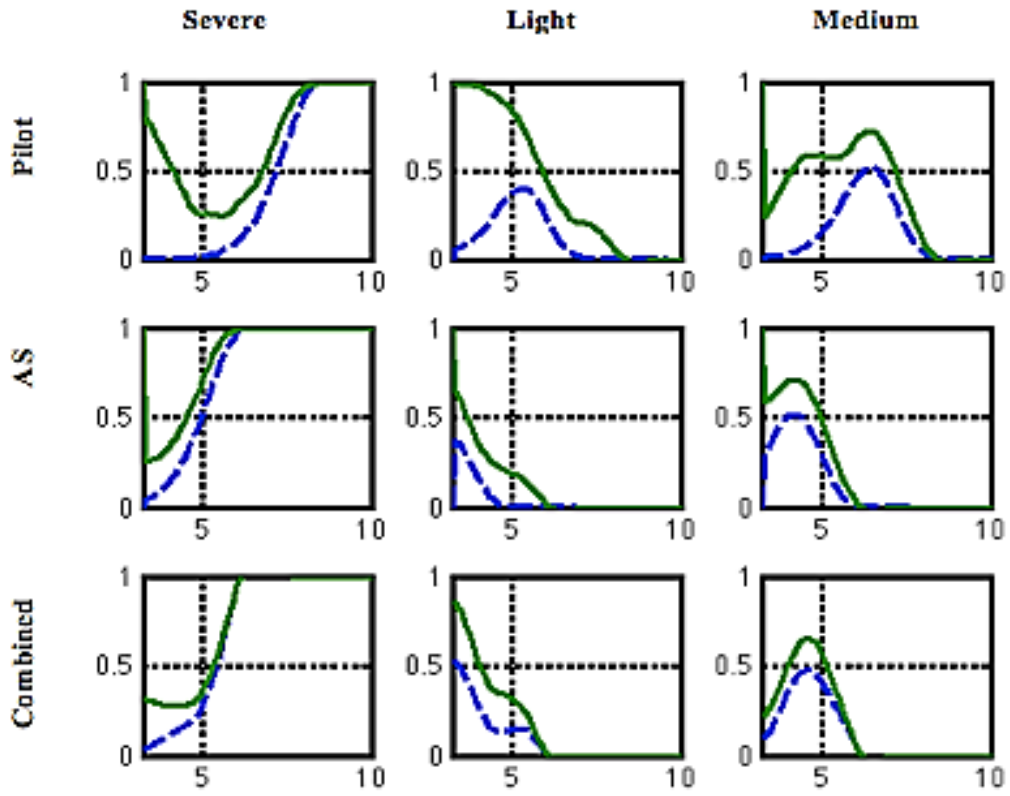


Figure 46. Probability estimated by the pilot, Automated System and the Combined Result

Table 5. MEB calculation for each result

MEB	Pilot estimated	AS estimated	Combined Result
Light	0.2237	0.0805	0.0480
Medium	0.3037	0.0842	0.0751
Severe	0.2152	0.0841	0.0486
Average	0.2475	0.0829	0.0572

10.3 Game Theory Result

First, we map the status evaluation to the action set based on the following table. Here, Action 1 stands for "continue flying" implying that no action is required. Action 2 stands for "prepare to land", which means that maintenance action must be taken after the vehicle reaches its destination. Action 3 stands for "land the aircraft immediately", which means that the aircraft's condition is severe and the pilot must land the vehicle immediately.

Since the automated system monitors the pilot's suggested action(s) automatically, it knows only what action the pilot is taking but not why he takes this particular action and its corresponding probability. Thus, the automated system will evaluate the current status and will estimate the corresponding probability. For example, we are to evaluate the risk for the automated system suggesting Action 1 but the pilot takes Action 3. There are four conditions that recommend Action 3 to be taken by the pilot:

Table 6. Probability for Each Condition

Condition	IGB	Oil cooler bearing	Probability
1	Faulty	Light	$\mathrm{Pr}_1 = \mathrm{p}_{32} \times \mathrm{p}_{21}$
2	Faulty	Medium	$\mathrm{Pr}_2 = \mathrm{p}_{32} \times \mathrm{p}_{22}$
3	Faulty	Severe	$\mathrm{Pr}_3 = \mathrm{p}_{32} \times \mathrm{p}_{23}$
4	Normal	Severe	$\mathrm{Pr}_4 = \mathrm{p}_{31} \times \mathrm{p}_{23}$

Then, referring to the risk table below:

Table 7. Risk Table for Taking Each Action

Components	Status	Risk for Action1	Risk for Action2	Risk for Action3
Oil cooler bearing Crack	Light	0	0	0
	Medium	16	0	0
	Severe	31	14	0
IGB	Normal	0	0	0
	Faulty	42	17	0

so the risk for taking Action 1 is:

$$R_{31} = \sum_{i=1}^{4} Pr_i \, r_i = 42Pr_1 + 58Pr_2 + 73Pr_3 + 31Pr_4$$

The cost corresponding to each action is estimated as follows:

Table 8. Cost Table for Taking Each Action

Action	Action 1	Action 2	Action 3
Cost	0	25	50

Cost for taking Action 1 is, of course, zero. The proposed formulation provides thus both cost and risk information. The pilot's suggested action and the AS's advisory is illustrated in Figure 47. Generally, the situation estimated by the automated system is more severe than that of the pilot. Thus, the action suggested by the automated system tends to cost more and is more likely to avoid some severe risks. The combined result, which is the optimum under the given payoff function, is shown in Figure 48.

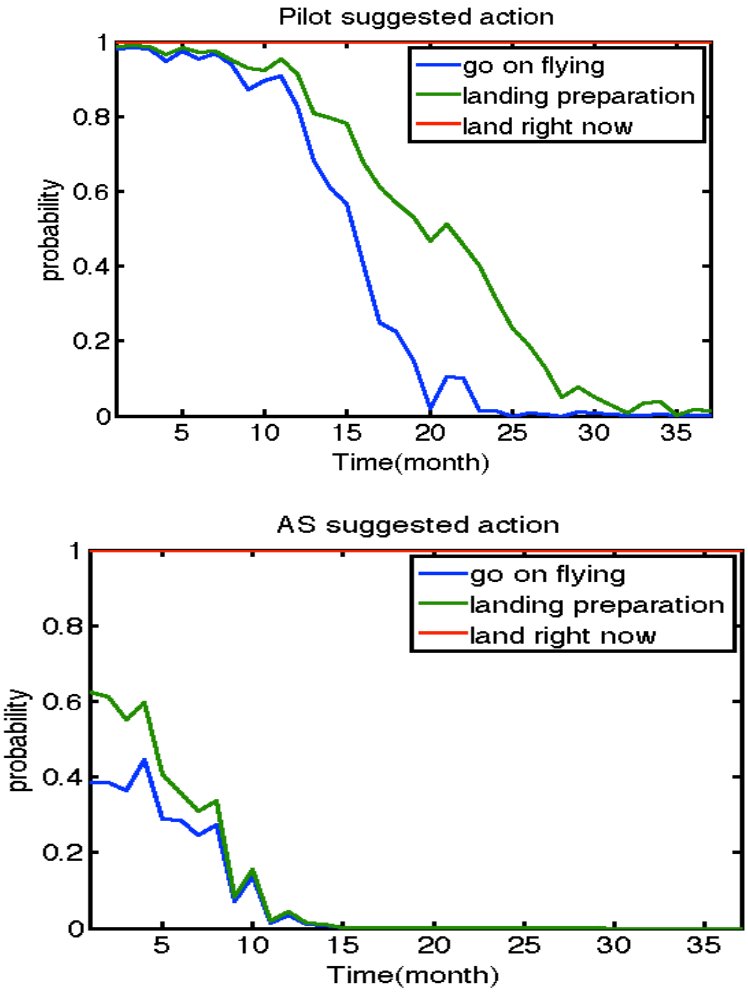


Figure 47. Suggested actions given by the pilot and Automated System

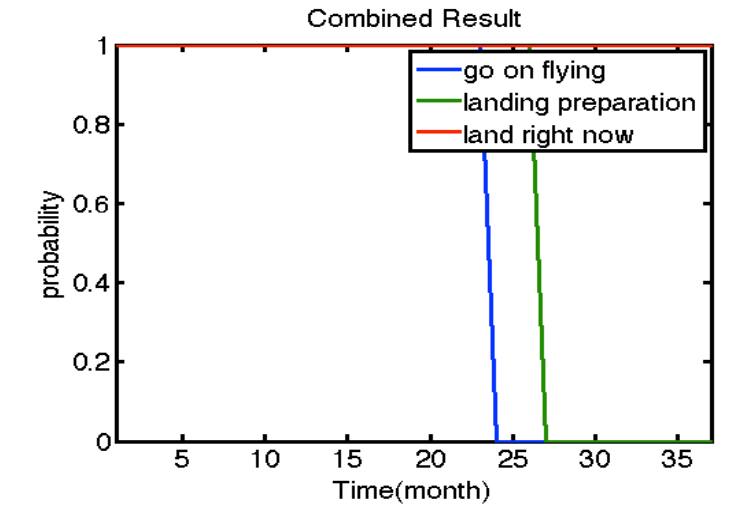


Figure 48. Combined Advisory

10.4 GUI Reasoning Result

We apply the GUI to demonstrate the dynamic behavior of the reasoning structure. At the beginning the system is inactive, corresponding to time 0 in the structure.

With the Start button pressed down, the system is activated and the Automated System's reasoning is initiated. At this moment, the pilot has not completed the reasoning phase and the knowledge base is in a hold state until the pilot's suggestion is generated and the probabilities of the estimated actions are shown as not available. This is the demonstration of the system condition between time t_0 and t_1, as shown in Figure 49.

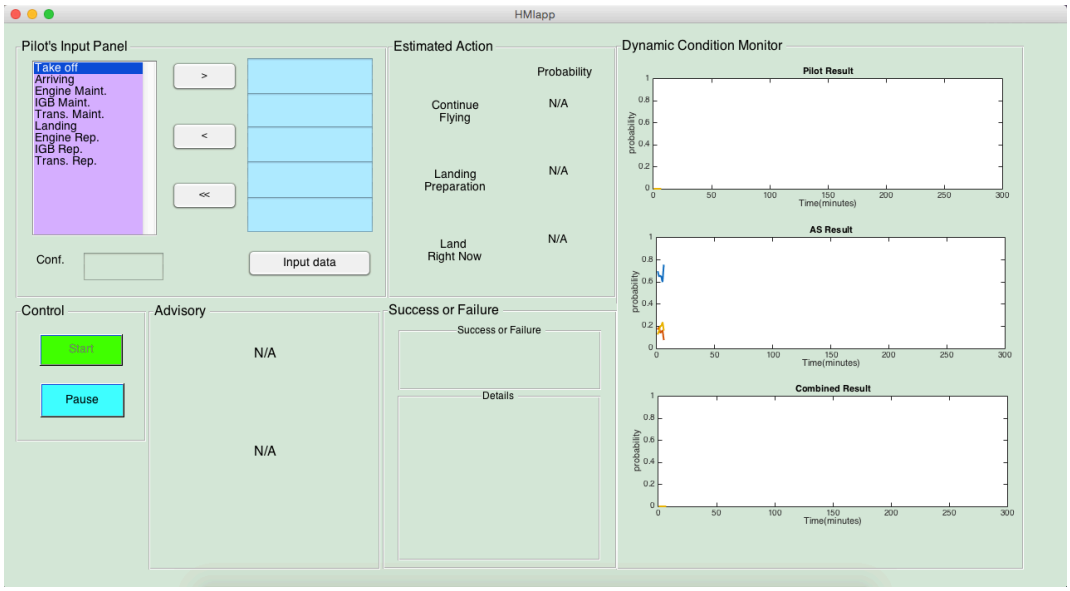


Figure 49. System Condition at time t_0

During the next phase, the pilot inputs the action sequences, as shown in Figure 50, and the probabilities associated with the estimated action sequence appear

indicating that the first task after time t_1, i.e. action estimation, has been accomplished. The knowledge base undertakes the task now to resolve the conflict between the automated system and the pilot, and estimate the final advisory. In the Combined Result figure, the advisory is changed to a pure strategy, rather than the mixed one. The final advisory is also shown in the Advisory section; the current state suggests that the advisory is Continue Flying. The Success or Failure module has completed the evaluation and believes at this moment that the actions are successful.

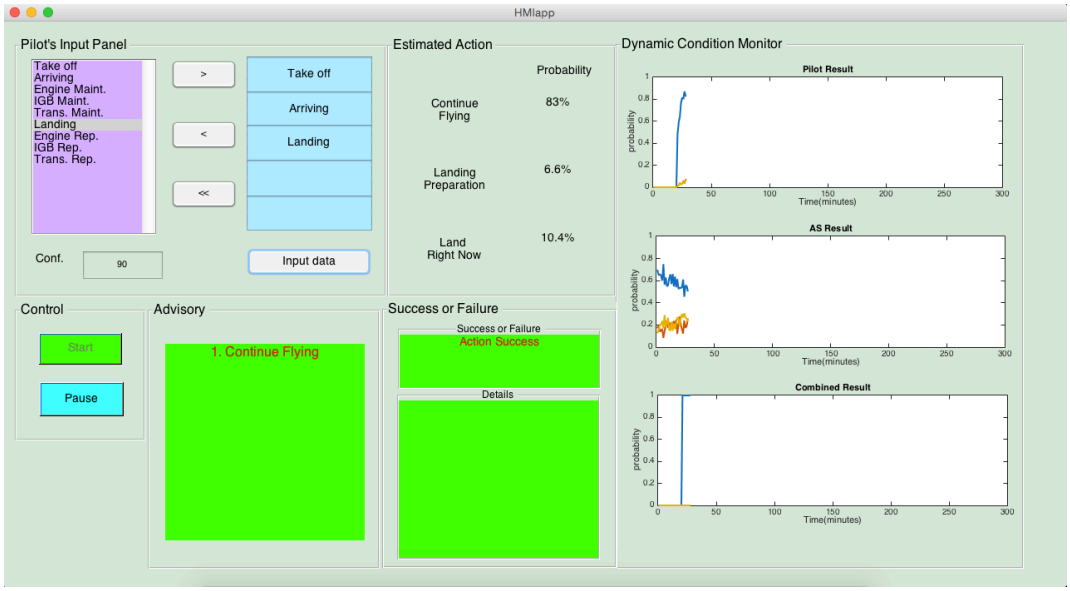


Figure 50. System Condition at time t_1

Figure 51 illustrates, as time progresses, when the action provided by the knowledge base is considered as failed. But the final advisory is not changed due to this indicated failure because the final advisory generation is decided by a combined consideration of the risk and cost. This failure is only shown to the pilot as additional information to assist the pilot to reconsider the sequence to be

executed.

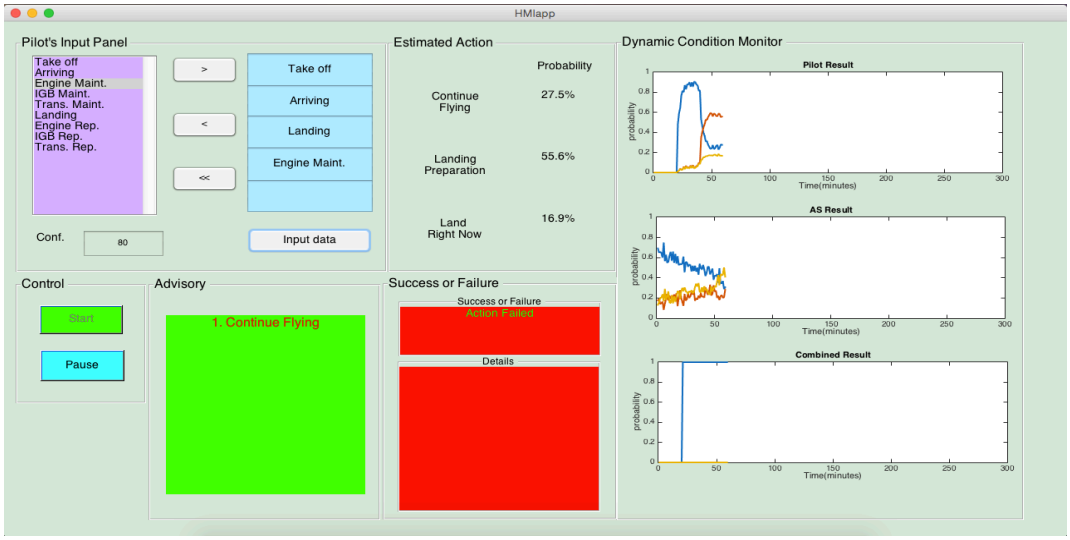


Figure 51. System Condition at time t_2

Then, the pilot takes more aggressive action, as illustrated in Figure 52. Thus, the combined result is changed to "Landing Preparation" and the new action provided is considered as a success.

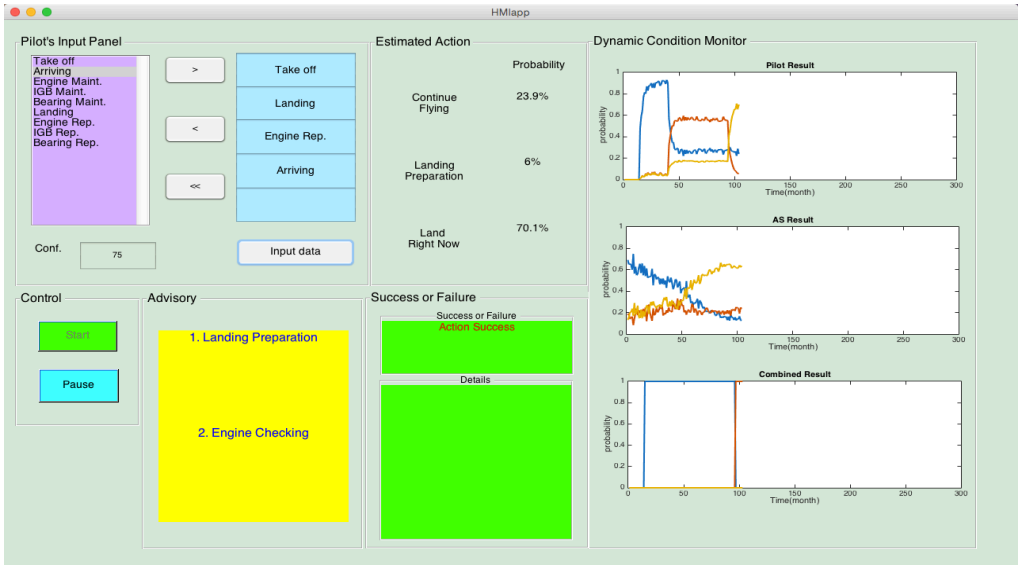


Figure 52. System Condition at time t_3

The final situation is illustrated in Figure 53. At this stage the advisory is “land the aircraft right now”. Because it is the most aggressive action the pilot can take, there is no need to analyze the success or failure state of the actions taken. Moreover, the knowledge base suggests also to the pilot to check the engine, in response to an indication that it is possible for the engine to be in a very severe condition disabling the aircraft to reach the nearest airport.

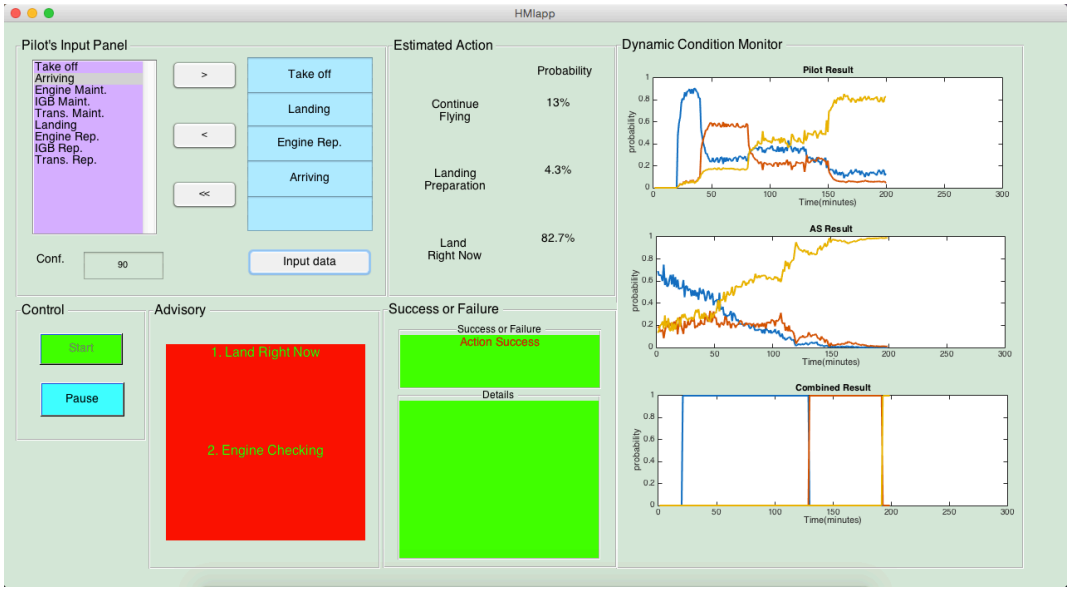


Figure 53. System Condition at time t_4

CHAPTER 11

CONCLUSIONS

Complex manned and unmanned aircraft require a harmonious synergy between the human operator and on-board automated processes for their safe, reliable and robust operation. Complexity issues and the lack of a rigorous human-machine interface have resulted over the past years in contingencies that is difficult or impossible to manage. This thesis introduces a system of systems approach to combine the human's observations, experience and expertise with on-board automated processes that monitor the health state of the aircraft and reason about the presence of detrimental fault/failure modes. The novel reasoning paradigm is employed to fuse evidences, resolve conflicts and provide to the human/pilot the "best" advisories to the pilot needed to mitigate the emergency. Such novel technologies are enabled by advances in sensing, computing and communications. These emerging tools/methods are in need of additional research to improve further the autonomy, reliability and resilience of critical assets.

11.1 Main Contribution

- Provided a methodology for human (pilot, operator)-machine (aircraft, process) interface technologies for improved machine performance and reduced operator workload

- A novel human-machine conflict resolution approach building on concepts from Dempster-Shafer Evidential theory and Game Theory intended to provide the "best" advisories to the operator for action in the event of machine contingencies (fault/failure modes)
- Defined reasonable performance metrics for the knowledge base to match new cases with the stored ones.
- Developed a MATLAB® Graphical User Interface (GUI) to demonstrate the effectiveness of our human-machine interface structure

11.2 Remaining Work

- More components and more fault modes can be further considered and generate a more complete demonstration
- Pick up suitable hardware to build the corresponding touchscreen panel and test the system.
- Redesign the interface in the cockpit and include the sections related to pilot's in the newly designed panel
- Use an aircraft application domain for proof of concept and validation purposes

REFERENCES

Basir, O., Yuanm, X., "Engine fault diagnosis based on multi-sensor information fusion using Dempster–Shafer evidence theory," in Information Fusion 8 (2007) 379–386

Bauer, M., "Approximation algorithms and decision making in the Dempster-Shafer theory of evidence—An empirical study," in International Journal of Approximate Reasoning 1997; 17: 217-237

Bloch, I., "Some aspects of Dempster-Shafer evidence theory for classification of multi-modality medical images taking partial volume effect into account", in Pattern Recognition Letters Volume 17, Issue 8, 1 July 1996, Pages 905–919

Bole, B., Tang., L., Goebel, K., Vachtsevanos, G., "Adaptive Load-Allocation for Prognossis-Based Risk Management", International Conference on Prognostics and Health Management, 2011.

Brown, D., Bole, B., and Vachtsevanos, G., "A prognostics enhanced reconfigurable control architecture," in 18th Mediterranean Conference on Control and Automation, pp. 1061-1066, June 23-35 2010.

Brown, D., Georgoulas, G., Bae, H., Chen, R., Ho, Y., Tannenbaum, G., Schroeder, J.B., and Vachtsevanos, G., "Particle filter based anomaly detection for aircraft actuator systems," in IEEE Aerospace Conference, March 7-14, 2009

Brown, D., Georgoulas, G., Bole, B., Pei, H.-L., Orchard, M., Tang, L., Saha, B., Saxena, A., Goebel, K., and Vachtsevanos, G., "Prognostics enhanced reconfigurable control of electro-mechanical actuators," in Proceedings of 2nd International Conference on Prognostics and Health Management (PHM), San Diego, CA, USA, September 27 - October 1 2009.

Brown, D., Georgoulas, G., Zhang, B., Edwards, D., Vachtsevanos, G., "Real-ime Fault Detection and Accommodation for Resolver Position Sensors," in Proceedings of 1st International Conference on Prognostics and Health Management 2008, PHM'08, Denver, CO, Oct.6-9, 2008.

Chen, C., Brown, D., Zhang, B., Vachtsevanos, G., Orchard, M., “A.NET Framework for an Integrated Fault Diagnosis and failure Prognosis Architecture”, in Proceedings of IEEEAUTOTESTCON, pp. 36-41, 2010.

Chen, C., Zhang, B., Vachtsevanos, G., “Prediction of Machine Health Condition Using Neuro-Fuzzy and Bayesian Algorithms” in Instrumentation and Measurement, IEEE Transactions on (Volume:61 , Issue: 2)

Dromigny, A., Zhu, Y., “Improving the dynamic range of real-time X-ray imaging systems via bayesian fusion”, in Journal of Nondestructive Evaluation September 1997, Volume 16, Issue 3, pp 147-160

Flemisch, F. O. “The Cognitive Assistant System and its Contribution to effective Man / Machine Interaction,” in the RTO SCI Symposium on “The Application of information Technologies (Computer Science) to Mission Systems”, held in Monterey, California, USA, 20-22 April 1998

Goebel K., Vachtsevanos G., “Algorithms and Their Impact on Integrated Vehicle Health Management”, in Integrated Vehicle Health Management, Perspectives on an Emerging Field, SAE International, 67-76, 2011

H´egarat-Mascle, S., Bloch, I., Vidal-Madjar, D., “Application of Dempster–Shafer Evidence Theory to Unsupervised Classification in Multisource Remote Sensing,” in IEEE Transactions On Geoscience And Remote Sensing, Vol. 35, No. 4, July 1997

James, M., Dubon, L., “An Autonomous Diagnostic and Prognostic Monitoring System for NASA's Deep Space Network”, in Aerospace Conference Proceedings, 2000 IEEE (Volume:2)

Kaftandjiana, V., Dupuisa, O., Babota, D., Zhub, Y. ” Uncertainty modelling using Dempster–Shafer theory for improving detection of weld defects”, in Pattern Recognition Letters Volume 24, Issues 1–3, January 2003, Pages 547–564

Lucas, P.,” Bayesian Networks in Medicine: a Model-based Approach to Medical Decision Making” in Proceedings of the EUNITE workshop on intelligent systems in patient care. Vienna, 2001

Nasir, A., Atkins, E.M., Kolmanovsky, I.V., “Conflict Resolution Algorithms For Fault Detection and Diagnosis” in Infotech@Aerospace 2011

Orchard M., Vachtsevanos G., Goebel K., “A Combined Model-Based and Data-Driven Prognostic Approach for Aircraft System Life Management”, in Machine Learning and Knowledge Discovery for Engineering Systems health management, CRC Press, 363-394, 2011

Orchard, M. and Vachtsevanos, G., “A Particle Filtering Approach for On-Line Fault Diagnosis and Failure Prognosis,” Transactions of the Institute of Measurement and Control, vol. 31, no. 3-4, pp. 221-246, June 2009.

Orchard, M., Tobar, F., and Vachtsevanos, G., “Outer Feedback Correction Loops in Particle Filtering-based Prognostic Algorithms: Statistical Performance Comparison,” Studies in Informatics and Control, Informatics and Control Publications, Volume 18, Issue 4, pp. 295-304, 2009.

Orchard, M., “A particle-filtering based framework for on-line fault diagnosis and failure prognosis. Ph.D Dissertation. School of Electrical and Computer Engineering,“ Georgia Institute of Technology. Atlanta, GA : s.n., 2007.

Rerbal, S., Benabdellah M., “Development of a Human Machine Interface of Information and Communication in telemedicine HMI-ICTM: Application to Physiological Digital Signal Processing in Telemedicine” in International Journal of Engineering Inventions, Volume 2, Issue 6 (April 2013) PP: 24-33

Rodrigues, M. A., Liu, Y., Bottaci, L., Rigas, D. I., ”Learning and Diagnosis in Manufacturing Processes through an Executable Bayesian Network”, in Intelligent Problem Solving. Methodologies and Approaches Lecture Notes in Computer Science Volume 1821, 2000, pp 390-396

Ross, T., Burnett, G.,” Evaluating the human–machine interface to vehicle navigation systems as an example of ubiquitous computing” in International Journal of Human-Computer Studies, Volume 55, Issue 4, October 2001, Pages 661–674

Saxena, A. " Knowledge-Based Architecture for Integrated Condition Based Maintenance of Engineering Systems" Ph.D. dissertation, Georgia Institute of Technology, 2007.

Shafer, G, A Mathematical Theory of Evidence, Princeton University Press, 1976,

Vachtsevanos G., Goebel K., “Basic Principles”, in Integrated Vehicle Health Management, Perspectives on an Emerging Field, SAE International, 55-66, 2011

Vachtsevanos, G., Kang, H. Cheng, J., and Kim, I., "On the Detection and Identification of Axial Flow Compressor Instabilities," American Institute of Aeronautics and Astronautics, Journal of Guidance, Control, and Dynamics, vol. 15, no. 5, pp. 1216-1223, 1992.

Vachtsevanos, G., Lewis, F., Roemer, M., Hess, A. and Wu, B., Intelligent Fault Diagnosis and Prognosis for Engineering Systems, John Wiley & Sons, Inc. 2006

Walsdorf, A., Putter, H., Onken, R.,“The cognitive process and its application within cockpit assistant systems” in Intelligent Transportation Systems, 1999. Proceedings. 1999 IEEE/IEEJ/JSAI International Conference on

Wu, B., Saxena, A., Patrick, R., and . Vachtsevanos, G., “Vibration monitoring for fault diagnosis of helicopter planetary gears.” In Proceedings of the 16th IFAC World Congress, 2005.

Yager, R.R., “On the Dempster-Shafer framework and new combination rules,” in Information Sciences 4, 93-137(1987)

Yang, B., Kim, K., “Application of Dempster–Shafer theory in fault diagnosis of induction motors using vibration and current signals”, in Mechanical Systems and Signal Processing Volume 20, Issue 2, February 2006, Pages 403–420

Zhang, B., Khawaja, T., Patrick, R., Vachtsevanos, G., Orchard, M., and Saxena, A., “A novel blind deconvolution de-noising scheme in failure prognosis,” Transactions of the Institute of Measurement and Control 32, 1, pp. 3-30, 2010.

Zhang, B., Khawaja, T., Patrick, R., Vachtsevanos, G., Orchard, M., and Saxena, A., “Application of Blind Deconvolution Denoising in Failure Prognosis”, IEEE Transactions on Instrumentation and Measurement, 58(2): 303–310, 2009.

C.A. Miller, J. Hamell, T. Barry, H. Ruff, M.H. Draper, and G. L. Calhoun, Adaptable Operator-Automation Interface for Future Unmanned Aerial Systems Control: Development of a Highly Flexible Delegation Concept Demonstration, Proceedings of Infotech@Aerospace Conference, American Institute of Aeronautics and Astronautics, Reston, VA (June 2012).

G. L. Calhoun, M.H. Draper, H.Ruff, T. Barry, C.A. Miller and J. Hamell, Future Unmanned Aerial Systems Control: Feedback on a Highly Flexible Operator-Automation Delegation Interface Concept, Proceedings of Infotech@Aerospace Conference, American Institute of Aeronautics and Astronautics, Reston, VA (June 2012).

K.B. Sullivan, K.M. Feigh, R. Mappus IV, F.T. Durso, U. Fischer, V. Pop, K.L. Mosier, D.G. Morrow, Using Neural Networks to Assess Flight Deck Human-Automation Interaction, Reliability Engineering & System Safety, 2013

F.T. Durso, E.J. Stearman, D.G. Morrow, K.L. Mosier, U. Fischer, V. Pop, and K.M. Feigh, Exploring Relationships of Human-Automation Interaction Consequences on Pilots: Uncovering Subsystems, Human Factors: The Journal of the Human Factors and Ergonomics Society, 2014

M.L Bolton, E.J. Bass, R.I. Siminiceanu, Using Formal Verification to Evaluate Human-Automation Interaction: A Review, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, VOL. 43, NO. 3, MAY 2013

S.B. Banks, C.S. Lizza, Wright-Patterson Air Force Base, Pilot's associate: a cooperative, knowledge base system application, IEEE Intelligent Systems and their Applications, June 1991 V6(3) pp 18-29

G. Calhoun, Pilot-Vehicle Interface, RTO-EN-12.